2021

Re-designing Main Memory Subsystems with Emerging Monolithic 3D (M3D) Integration and Phase Change Memory Technologies

Chao-Hsuan Huang
University of Kentucky, michael987852@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0002-6909-9729
Digital Object Identifier: https://doi.org/10.13023/etd.2021.450

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Chao-Hsuan Huang, Student

Dr. Ishan G. Thakkar, Major Professor

Dr. Daniel Lau, Director of Graduate Studies

Re-designing Main Memory Subsystems with Emerging Monolithic 3D (M3D) Integration and Phase Change Memory Technologies

THESIS

A thesis submitted in partial
fulfillment of the requirements for
the degree of Master of Science in
Electrical Engineering in the College
of Engineering at the University of
Kentucky

By
Chao-Hsuan Huang
Lexington, Kentucky

Director: Dr. Ishan G Thakkar, Assistant Professor of Electrical Engineering
Lexington, Kentucky
2021

https://orcid.org/0000-0002-6909-9729

ABSTRACT OF THESIS

Re-designing Main Memory Subsystems with Emerging Monolithic 3D (M3D) Integration and Phase Change Memory Technologies

Over the past two decades, Dynamic Random-Access Memory (DRAM) has emerged as the dominant technology for implementing the main memory subsystems of all types of computing systems. However, inferring from several recent trends, computer architects in both the industry and academia have widely accepted that the density (memory capacity per chip area) and latency of DRAM based main memory subsystems cannot sufficiently scale in the future to meet the requirements of future data-centric workloads related to Artificial Intelligence (AI), Big Data, and Internet-of-Things (IoT). In fact, the achievable density and access latency in main memory subsystems presents a very fundamental trade-off. Pushing for a higher density inevitably increases access latency, and pushing for a reduced access latency often leads to a decreased density. This trade-off is so fundamental in DRAM based main memory subsystems that merely looking to re-architect DRAM subsystems cannot improve this trade-off, unless disruptive technological advancements are realized for implementing main memory subsystems.

In this thesis, we focus on two key contributions to overcome the density (represented as the total chip area for the given capacity) and access latency related challenges in main memory subsystems. *First*, we show that the fundamental area-latency trade-offs in DRAM can be significantly improved by redesigning the DRAM cell-array structure using the emerging monolithic 3D (M3D) integration technology. A DRAM bank structure can be split across two or more M3D-integrated tiers on the same DRAM chip, to consequently be able to significantly reduce the total on-chip area occupancy of the DRAM bank and its access peripherals. This approach is fundamentally different from the well known approach of through-silicon vias (TSVs)-based 3D stacking of DRAM tiers. This is because the M3D integration based approach does not require a separate DRAM chip per tier, whereas the 3D-stacking based approach does. Our evaluation results for PARSEC benchmarks show that our designed M3D DRAM cellarray organizations can yield up to 9.56% less latency and up to 21.21% less energy-delay product (EDP), with up to 14% less DRAM die area, compared to the conventional 2D DDR4 DRAM. *Second*, we demonstrate a pathway for eliminating the write disturbance errors in single-level-cell

PCM, thereby positioning the PCM technology, which has inherently more relaxed density and latency trade-off compared to DRAM, as a more viable option for replacing the DRAM technology. We introduce low-temperature partial-RESET operations for writing '0's in PCM cells. Compared to traditional operations that write '0's in PCM cells, partial-RESET operations do not cause disturbance errors in neighboring cells during PCM writes.

The overarching theme that connects the two individual contributions into this single thesis is the density versus latency argument. The existing PCM technology has 3 to 4× higher write latency compared to DRAM; nevertheless, the existing PCM technology can store 2 to 4 bits in a single cell compared to one bit per cell storage capacity of DRAM. Therefore, unlike DRAM, it becomes possible to increase the density of PCM without consequently increasing PCM latency. In other words, PCM exhibits inherently improved (more relaxed) density and latency trade-off. *Thus*, both of our contributions in this thesis, the first contribution of re-designing DRAM with M3D integration technology and the second contribution of making the PCM technology a more viable replacement of DRAM by eliminating the write disturbance errors in PCM, connect to the common overarching goal of improving the density and latency trade-off in main memory subsystems. *In addition*, we also discuss in this thesis possible future research directions that are aimed at extending the impacts of our proposed ideas so that they can transform the performance of main memory subsystems of the future.

KEYWORDS: DRAM, Monolithic 3D Integration, Phase Change Memory, Emerging, Partial-Reset, DRAM Access latency

Chao-Hsuan Huang

December 16, 2021

Re-designing Main Memory Subsystems with Emerging Monolithic 3D (M3D) Integration and Phase Change Memory Technologies

By
Chao-Hsuan Huang

Dr. Ishan G Thakkar
Director of Thesis

Dr. Daniel Lau
Director of Graduate Studies

December 16, 2021
Date

ACKNOWLEDGMENTS

I would like to thank the committee members for taking their time to provide me valuable feedback. This thesis would not exist if not for my advisor Dr. Thakkar's help. I still remember the time when I was new to the school and not knowing what to do, where to start. That is when Dr. Thakkar gave me support and direction not only for research but also for other important aspects of life. I feel like I still have a lot to learn from him and I do still have a lot to improve upon. My very first conference IEEE/ACM ESWEEK 2019, was the most fun I had. I think I did alright on the presentation for my research paper. However, when I was up there trying to present my lab mate's paper, I was not that well-prepared. At that time, Dr. Thakkar encouraged me. Thank you once again for helping me in going through my Master's degree.

I would like to also thank my lab mates, Praneeth, Sairam and, Surpreeth. Praneeth, it was really fun to work with you guys. Praneeth and Sairam, you guys helped me a lot during the COVID time. I was so lonely and afraid during that time, but you guys always called me frequently to make sure I was okay. Surpreeth, you always gave me good tips on Linux systems. I will miss the time you guys made fun of me. I wish that we could attend another in-person conference again together.

My mom, and my dad, you may never read this but I love you, and thank you for giving me this opportunity to study abroad. You are very supportive parents and don't worry about me getting a job. I will find one eventually!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Motivation

Main memory is a very common subsystem that is found in all types of computers. Here, the term 'computers' encompasses all types and scales of devices and systems that can perform electrical computation and information processing. Main memory in computers supports multitasking by users by providing fast access to switch between applications. The access speed of the main memory lays between the low-latency last-level cache and the high-latency hard disk. It has been shown that the emerging latency-critical, data-centric workloads related to Artificial Intelligence (AI), Big Data, and Internet-of-Things (IoT) spend more than 50-60% of their total execution time accessing the main memory subsystems [25]. Moreover, main memory subsystems are found to be consuming close to 40% of the total system energy [25]. Therefore, improving the latency and energy behaviour of main memory subsystems can have substantial impact on the overall performance of entire computing systems.

The dominant main memory technology is Dynamic Random-Access Memory (DRAM) today, which has a few shortcomings. First, an individual DRAM cell consists of one transistor and one capacitor (1T1C). One DRAM cell stores only one bit (logic '0' or '1') of information, which harms the density of DRAM compared to other technologies such as flash and Phase Change Memory (PCM) that can store multiple bits per cell. Second, due to the volatile nature of DRAM, DRAM cells lose information with time as the charge in the cell capacitors leaks out with time. To counter this, DRAM based main memories have to spend extravagant amount of time and energy for data retention by performing frequent refresh operations. During refresh operations, DRAM remains unavailable for regular read/write access requests, which seriously impacts the performance and energy-efficiency of DRAM subsystems. Third, it has been observed that the DRAM latency has not scaled as much as the DRAM density. Over the past 20 years, the DRAM cell density has improved by 128×, whereas the latency has only improved by 30%. This problem is referred to as "Memory Wall" [94] [53] [3]. The reason for the Memory Wall problem is two-fold: (1) the fundamental latency-density trade-off of DRAM technology that makes it very difficult to simultaneously improve both the DRAM latency and density; and (2) the cost-centric mindset of DRAM industry that has preferred DRAM density to DRAM latency through the years DRAM evolution. Inferring from these shortcomings, computer architects in both the industry and academia have widely accepted that the main memory density and latency cannot sufficiently scale in the future to meet the requirements of future workloads, unless disruptive technological advancements for implementing main memory subsystems are realized.

To address these shortcomings, researchers from the industry and academia have explored two pathways: (1) Re-designing DRAM cell-array organization to improve the density-latency trade-off of DRAM; (2) Aiming to replace DRAM technology with more scalable PCM technology. A myriad of prior works have looked to re-architect

DRAM subsystems to address the high access latency problem. But none of these works has been able to notably improve the density-latency trade-off. Because the density-latency trade-off is very inherent to the memory technology. On the other hand, researchers from the industry and academia have also recently explored PCM as a potential technology that can replace DRAM. PCM based main memory can present inherently improved density-latency trade-offs compared to DRAM based main memory, because each physical PCM cell can store multiple bits of information compared to a DRAM cell. Moreover, PCM can also provide non-volatile, persistent storage, which is contrary to DRAM, saving PCM based main memory subsystems from spending extravagant amount of energy and time for data retention. Despite these benefits, however, PCM based main memory implementations still face several daunting challenges such as high write latency, low write endurance and reliability, and write disturbance errors. Because of these challenges, although several PCM prototypes have emerged over the past few years, PCM has not clearly replaced DRAM in a widespread manner yet.

In this thesis, we focus on two key contributions to overcome the density (represented as the total chip area for the given capacity) and access latency related challenges in main memory subsystems. Our contributions are detailed in Chapters 2 and 3, and summarized in Section 1.3. In the next section, a brief review of the most relevant related works that looked to address the shortcomings of DRAM and PCM based main memory subsystems is presented.

1.2 Related Work

Tier-Latency DRAM [53] address on improving access speed by separating the bitline into the short and long segment with minimum area cost. The Shorter bitline reduce the critical path thus improve the fundamental access latency. Refresh drain a lot of energy over time, more even as the capacity of the DRAM grows. Memory bank could not be accessed while refreshing. A better refresh mechanism like RAIDR [60] can solve this problem. It avoids refresh operation in subarrays that have just been accessed to save energy. 3D stacked DRAM [86] [35] stacked DRAM dies together using though silicon vias(TSVs) to exploit parallelism and improve bandwidth. As for scalability, emerging memories could be the solution, phase change memory provides better scalability because it can store multiple bits in one cell.

The challenge for Tier-Latency DRAM, however, is that you need a way to determine what should be store in a short segment bitline and does not work if heavy load is required in specific applications. Refresh mechanism is generally good but requires the memory controller to control refresh dynamically. 3D stacked DRAM hit the limit as the TSVs could take up large space and the yield problem still needs to be overcome. Phase change memory is generally slow compared to DRAM. The change of its resistance requires heat transfer and error correction to ensure data reliability which increases latency and energy consumption.

1.2.1 Prior Work on DRAM Performance Enhancement

Short Bitlines. Some specialized DRAMs reduce access latency by reducing the number of cells-per-bitline, e.g. Micron's RLDRAM [89] and Fujitsu's FCRAM [77]. However, this requires the addition of more sense amplifiers, incurring an area overhead of 30–80% [42] [77]. This results in significantly higher cost-per-bit.

Cached DRAM. Cached DRAM [1, 2, 26, 27, 29, 41, 76, 93, 102] adds an SRAM cache to DRAM chips. However, SRAM-cached DRAM approaches have two major limitations. First, an SRAM cache incurs significant area overhead; using CACTID [88], the estimation is that an SRAM-cached DRAM with equivalent capacity to a TL-DRAM with 32 rows/near segment would incur 145.3% area overhead. Second, transferring data between the DRAM array and the SRAM cache requires use of the relatively narrow global I/O bus within the DRAM chip leading to high caching latency. In contrast, TL-DRAM incurs minimal area overhead of 3.15% and facilitates fast in-DRAM transfer of data between segments. With the same amount of cache capacity, the performance improvement of Cached DRAM (8.3%) is less compared to that of TL-DRAM (12.8%) for 1-core systems. This is primarily due to the large caching latency incurred by Cached DRAM.

Increased DRAM Parallelism. Kim et al. [45] propose schemes to parallelize accesses to different subarrays within a bank, thereby overlapping their latencies. Multiple works [6, 92, 103] have proposed partitioning a DRAM rank into multiple independent rank subsets that can be accessed in parallel [5]. All of these proposals reduce the frequency of row-buffer conflicts, but not their latency. DRAM Controller Optimizations. Sudan et al. [83] propose a mechanism to co-locate heavily reused data in the same row, with the goal of improving row-buffer locality. A large body of prior work has explored DRAM access scheduling in the controller (e.g. [13,23,43,44,62,63]).

Segmented Bitlines in Other Technologies. Prior works [24, 75, 80] have proposed the use of segmented bitlines in other memory technologies, like SRAM caches and Wash memory. In contrast, this is, to our knowledge, the first work to propose the use of segmented bitlines in DRAM. Our approach and the resulting tradeoffs are different as we take into account characteristics and operation that are unique to DRAM, such as DRAM-specific subarray organization, sensing mechanisms, and timing constraints.

1.2.2 Prior Work on PCM Optimizations

In recent years, a compelling body of research has been conducted that aims to minimize the effect of longer write latency on PCM performance [21,36,39,40,46,48,57, 59,64,69,71,97,98], [4,54,55,100]. The PCM latency improving methods presented in the literature can be broadly classified into the following four categories: 1) methods that optimize DRAM-PCM hybrid memory architectures to reduce the number of write operations to PCM [4, 54, 55, 100]; 2) methods that hide longer write latency by scheduling PCM writes among idle bank cycles [46,48,69,71]; 3) architecture-level solutions for reducing write latency in MLC PCMs [36,39,40,64,97]; and 4) methods that utilize latency-aware data coding schemes to relax the need for writing logic 1

bits in some write operations, thereby reducing the average write latency [21, 59, 98].

The prior works (e.g., [4, 54, 55, 100]) that utilize DRAM–PCM hybrid memory systems, in general, optimize the memory space and page allocation between the DRAM and PCM parts of the main memory to reduce the number of write operations to the high-latency PCM part. Lee et al. [54] presented one of the earliest works on DRAM–PCM hybrid memory, which provides a comprehensive design space exploration of DRAM–PCM hybrid memory systems from the perspective of energy-delay efficiency. Khouzani et al. [4] utilized DRAM as a cache to PCM and propose a DRAM page replacement algorithm along with a conflict-aware page remapping strategy to reduce the number of DRAM misses and the resultant write backs to PCM. Lee et al. [55] proposed a write-history aware page replacement algorithm for hybrid DRAM–PCM architectures that estimates future write references based on write history, and then absorbs frequent writes into DRAM. Zhang et al. [100] proposed a write-back aware last-level cache management scheme for the hybrid DRAM–PCM main memory, which improves the cache hit ratio of PCM blocks and minimizes write-backs to PCM.

A higher write latency can be masked using a DRAM–PCM hybrid memory system with intelligent page allocation and scheduling as long as there is sufficient write bandwidth [71]. Thus, a DRAM–PCM hybrid memory system draws forth an untrue masked behavior of the PCM subsystem, as the DRAM part of the hybrid system hides the longer write latency of the constituent PCM part. However, as explained in [71], the inherently longer latency of PCM write-back accesses, due to DRAM misses, may stall subsequent read accesses, significantly increasing average read latency of the hybrid system. As read accesses are latency critical, increasing read latency has significant performance impact [71]. Therefore, improving the true unmasked write latency of PCM accesses is imperative for improving the overall memory performance.

To reduce the unmasked write latency of PCM, some prior works (e.g., [46, 48, 69, 71]) tend to schedule write requests during idle bank cycles when the target banks are not serving any other requests. Qureshi et al. [71] presented write pre-emption that pre-empts the on-going write operation to serve a newly arrived read request to the same bank, thereby reducing the effect of longer write latencies on average read latency of the PCM system. Kim et al. [46] proposed to overlap the resistance drift latency of some write operations with concurrent read operations, thereby achieving significant benefits in overall system performance. Qureshi et al. [69] exploited the property of PCM devices where PCM write latency is longer than read latency only because of high-latency SET operations. They propose an architectural technique called PreSET that proactively SETs all the bits in a given memory line during idle bank cycles as soon as the line becomes dirty in the cache. Thus, subsequent write operations to the line require only RESET operations, which incur much lower latency. These write scheduling methods are efficient in hiding longer write latency, but cannot reduce the fundamental latency of every write access.

Some other architecture-level solutions (e.g., [36, 39, 40, 64, 97]) have been proposed for reducing write latency in MLC PCMs. These techniques are specific to MLC PCMs and they are not general enough to be applicable for SLC PCMs. An MLC

PCM, which stores multiple bits (represented by multilevel resistance) in a single cell, offers high density with low per-byte fabrication cost [40]. However, due to cell process variations and the relatively small differences among resistance levels, MLC PCM typically employs an iterative write scheme to achieve precise control, which suffers from large write access latency [40]. Moreover, the susceptibility to variations renders MLC PCM less reliable than SLC PCM, making it less preferable over SLC PCM. Therefore, we focus on optimizing SLC PCM in this paper.

Some other techniques (e.g., [21, 59, 98]) have been proposed that utilize latency-aware coding schemes to encode data words, which relax the need to write 1 bits in some write operations, resulting in a reduced average write latency. Cho and Lee [21] proposed Flip-N-Write, which on every write request, updates only those bits of the new data word that differ from the original data word. Flip-N-Write also limits the required number of bit updates to half of the data word size by "flipping" (inverting) the bit values of the new data word if the number of to-be-updated bits is over half the data word size. As a result, Flip-N-Write can achieve 2× write bandwidth by doubling the write unit size without increasing the instantaneous write current. Yue and Zhu [98] exploited a property of PCM cells that writing a 1 (SET operation) takes longer time but a smaller amplitude current than writing a 0 (RESET operation). They propose two-stage write, wherein a write is divided into two stages: 1) in the write-0 stage, all zeros are written at an accelerated speed and 2) in the write-l stage, all ones are written with increased parallelism, without violating power constraints. Two-stage write achieves better resource utilization and reduces the service time of writing a cache line. Li and Mohanram [59] proposed write-once-memory (WOM) code PCM architecture (WOMC_PCM), wherein they encode the PCM data words using a 2/3 WOM-code. A "t /n WOM-code" is a coding scheme that uses n "write-once bits" to represent one of v values so that the WOM can be written a total of t times by using only RESET operations. Therefore, 2/3 WOM-code used in WOMC_PCM [59] utilizes a 3-bit code to represent one of four two-bit values that can be written a total of two times by using only RESET operations. Thus, WOMC_PCM architecture reduces the necessity of using SET operations (to write 1s) during some writes, thereby reducing the latency for those writes resulting in significantly reduced average write latency. However, these methods (FlipN-Write [21], Two-stage write [98], and WOMC_PCM [59]) cannot eliminate the need to write 1 bits (the need to use SET operations) in every write operation, thus, limiting the achievable improvement in average write latency.

Read-While-Write in PCM : A patent application [15] describes read-while-write for PCM, where a read and write request can be scheduled simultaneously from a PCM bank using different partitions. However, no architectural technique is described on how to leverage this feature for system performance. Some earlier works such as [104] address architectural aspects assuming unrealistic system settings (such as infinite memory channel bandwidth). Our work not only addresses limitations of these prior works to resolve read-write bank conflicts, but also resolves read-read bank conflicts for the first time. We also evaluate PALP against a realistic version of [104] and find that PALP improves average system performance by 28%.

Performance/energy/endurance improvement of PCM :Many prior works

optimize performance and energy of PCM [12, 22, 50, 51, 70, 74, 96]. Cho et al. propose Flip-N-Write [22] to improve PCM performance by first reading the memory content and then programming only the bits that need to be altered. Qureshi et al. propose PreSET [70], an architectural technique that SETs the PCM cells of a memory location in the background before programming them during write. This improves performance by converting a write operation to a RESET operation of the PCM cells, which is faster. There are also techniques to consolidate multiple write operations [95] to reduce the number of cells that need to be programmed, saving energy and improving performance. To mitigate PCM's cell-level endurance problem, several wear-leveling techniques are proposed [7, 78]. PALP can be combined with these and similar techniques.

Writeback optimization : Several prior works propose line-level writeback [49–51, 67, 68, 74], where for each evicted DRAM cache block, processor cache blocks that become dirty are tracked and selectively written back to PCM. Various works propose dynamic write consolidation [52, 79, 82, 91, 95], where PCM writes to the same row are consolidated into one write operation. Other works propose write activity reduction [30, 33], where registers are allocated on CPUs to reduce costly write operations in PCM. Yet some other works propose multi-stage write operations [99, 101], where a write request is served in several steps rather than in one-shot to improve performance. Qureshi et al. propose a morphable PCM system [73], which dynamically adapts between high-density and high-latency MLC PCM and low-density and low-latency single-level cell PCM. Qureshi et al. propose write cancellation and pausing [72], which allows PCM reads to be serviced faster by interrupting long PCM writes. Jiang et al. propose write truncation [37], where a write operation is truncated to allow read operations, compensating for the loss in data integrity with stronger ECC. PALP is complementary to all these approaches.

Multilevel Cell PCM Optimizations : PCM cells can be used to store multiple bits per cell (referred to as multilevel cell or MLC). MLC PCM offers greater capacity per bit at the cost of asymmetric energy and latency in accessing the bits in a cell. Yoon et al. propose an architectural technique for data placement in MLC PCM [96], exploiting energy-latency asymmetries.

Review of Prior Works that Address Low Write Endurance of PCM

At architectural level, several solutions have been proposed to address the problem of write disturbance (WD) in PCM. Most of the conventional techniques build on a verify and correct (VnC) method in which the failed cells are rewritten after verification. Various schemes have been introduced to improve the system performance by reducing the frequency of a VnC operation. The proposed method eliminates WD within a word-line and can be used along with other orthogonal techniques that address WD across the bitlines.

[38] proposes a Data Insulation (DIN) technique, based on data compression and encoding to address WD in a wordline. In general, adding redundant bits to the data allows bit patterns that are not vulnerable to WD to be selected. DIN first applies compression to make room for redundant bits and then encodes the compressed data

to minimize the frequency of WD-vulnerable patterns. Finally, it adds a 2-bit error correcting code (BCH-2) which eliminates the need to perform a rewrite operation if two or less bits fail as a result of WD.

Tavana et al. [85] introduced two techniques to address WD along a word-line. The first one, called Data Modification with Partitioning (DMPart), divides an array into smaller words or partitions and counts the four 2-bit data patterns. Based on the pattern counts, DMPart applies an XOR-based encoding to minimize the frequency of WDvulnerable patterns. This method requires auxiliary bits (for decoding) to be stored along with the data bits, which somewhat reduces the effective storage density (data bits/cell) of the memory. To avoid extra storage overhead for auxiliary bits, the authors suggest to employ the unused Error Correcting Pointers (ECP) which have been proposed by [11] to address hard errors in PCM. This, however, may not be attractive when more and more ECP entries are used to address the hard errors. The second technique proposed in [85], called Selective Writes to Exposed Cells (SWEX), is based on the idea of trading lifetime to reduce the VnC overhead. In general, all the exposed bits (vulnerable to WD), including those that can fail in subsequent writes due to a domino effect, can be identified prior to a write. Writing all the exposed bits together eliminates the need for a VnC operation. This, however, considerably increases the bit flips and reduces the memory lifetime. SWEX is flexible as it writes to all the exposed bits only when the number of exposed bits is below a certain threshold (chosen to be 32 in [85]). This helps to achieve a compromise between lifetime reduction and performance improvement.

Wang et al. [90] introduce three (VnC-based) techniques to address WD across the bitlines. The first one is to make use of ECP. Unused ECP entries can be used to record the location of bits failing due to WD and VnC is performed only if there are more errors than ECP can handle. The second technique proposed in [90] is to read the adjacent word-lines when the line to be written is still in the write queue, thus reducing the overhead of a verify operation. Lastly, the authors propose to disable certain rows from being written when lowering the memory capacity is possible, thus avoiding WD to these rows as a result of a write to their adjacent rows.

A write scheme that exploits the latency imbalance between different cell groups of PCM is introduced in [9]. In this scheme, ECP entries are used to handle WD in those cell groups which have higher latencies while VnC is used for the cell groups with lower latencies. A hot page remapping by exploiting the temporal locality in memory accesses, is introduced in [10]. The writeintensive pages can be remapped to a WD-free region, thus mitigating the VnC overhead.

Eslami et al. [8] suggest a checkerboard layout and write-once-memory (WOM) code to address WD. A checkerboard layout eliminates WD both across the word-lines and bitlines, however, it allows only half of the full memory capacity. [84] proposes compression to reduce the number of written bits and store the data in left/right aligned format for even/odd wordlines. This reduces the possibility of WD across the bitlines. Additionally, the method introduced in [84] skips a VnC operation for adjacent lines if the lines are in the last-level cache.

In this chapter, we focus on eliminating the WD in a wordline, assuming that the WD across the bitlines is eliminated by either adding extra space or using other

orthogonal (VnC-based) approaches such as use of ECP [90], early read [90], disabling certain rows [90], remapping hot pages [10] and skipping VnC based on cached data [84]. Moreover, our evaluation shows that bitline WD mitigation techniques can be more efficient when the proposed encoding is used to eliminate WD within a wordline. We compare our work with the relevant methods such as a basic VnC scheme, DIN [38], DMPart [85] and SWEX [85] which also focus on addressing WD in a wordline.

1.3 Summary of Contributions

In this thesis, we have made two key contributions, which are summarized below.

- Over the years, the DRAM latency has not scaled proportionally with its density due to the cost-centric mindset of the DRAM industry. Prior work has shown that this shortcoming can be overcome by reducing the critical length of DRAM access path. However, doing so decreases DRAM area-efficiency, exacerbating the latency-area tradeoffs for DRAM design. In this contribution, we show that reorganizing DRAM cell-arrays using the emerging monolithic 3D (M3D) integration technology can improve these fundamental latency-area tradeoffs. Based on our evaluation results for PARSEC benchmarks, our designed M3D DRAM cellarray organizations can yield up to 9.56% less latency and up to 21.21% less energy-delay product (EDP), with up to 14% less DRAM die area, compared to the conventional 2D DDR4 DRAM [32].

- Phase Change Memory (PCM) is seen as a potential candidate that can replace DRAM as main memory, due to its better scalability. However, writing '0s' in PCM cells requires hightemperature RESET operations, which induce write disturbance errors in neighboring idle PCM cells due to excessive heat dissipation. This contribution introduces low-temperature partial-RESET operations for writing '0s' in PCM cells. Compared to traditional RESET operations, partial-RESET operations dissipate negligible heat, and therefore, do not cause disturbance errors in neighboring cells during PCM writes [31].

Chapter 2 Improving the Latency-Area Tradeoffs for DRAM Design with Coarse-Grained Monolithic 3D (M3D) Integration

2.1 Introduction

Over the years since the emergence of DRAM, various manufacturers have deliberately sacrificed the access latency benefits of the continuing DRAM process scaling, to achieve greater cell density (i.e., more DRAM cells per unit die area) and lower cost-per-bit for DRAM, by sharing area-hungry DRAM access peripherals (e.g., sense amplifiers (SAs)) with increasingly large number DRAM cells [53]. Consequently, most DRAM designs today have very long internal critical access path, corresponding to having many DRAM cells inter-connected through a long wire called a bitline [53]. This design choice has slowed down the DRAM latency scaling, which in turn has exacerbated the "Memory Wall" problem by widen-ing the performance gap between the processor and DRAM subsystems even further. To alleviate the "Memory Wall" problem, which is crucial for meeting the performance demands of the modern data-driven computing applications, an efficient solution has been to use short-bitline DRAM architectures (e.g., [47], [89], [77]). However, these architectures require more SAs for a given die capacity, increasing the die area, and thus, reducing the die's cell density and cost-per-bit. As a result of this inherent area-latency tradeoffs in short-bitline DRAMs, the industry has relegated them to specialized applications only such as high-end networking systems (e.g., [89]) that can tolerate a very high cost for a very low latency. For more widespread adoption of the short-bitline DRAM architectures, the per-die cell density for such DRAM architectures needs to be increased, for which improving the fundamental latency-area tradeoffs for DRAM design is of paramount importance. To improve the latency-area tradeoffs for DRAM design, and consequently improve the per-die cell density for DRAM, we propose to use the emerging monolithic 3D (M3D) integration technology [16]. In this chapter, we show for the first time that reorganizing the traditional 1T1C (1-transistor 1-capacitor) DRAM die (we consider DDR4 DRAM [7]) at the subarray-level granularity with the M3D technology can miti-gate the inherent latency-area tradeoffs for DRAM design, in spite of suffering from performance degradation related to the M3D fabrication process [65]. Our idea is to partition the sense-amplifiers and other peripherals on a different M3D tier from the tier with DRAM cell-arrays. We present two different M3D DDR4 DRAM designs, both with improved cell density (die area) and access latency, compared to the baseline 2D DDR4 DRAM of the same capacity. Our key contributions in this chapter are summarized below.

- To relax the latency-area tradeoffs for DRAMs, we reorganize the cell-array of the commodity 2D DDR4 DRAM [7] using the coarse-grained M3D integration technology;
- We present the subarray-level bank layouts as well as the latency, area, and energy analysis (based on SPICE and other circuit-level simulations) for our

designed M3D DDR4 DRAMs;

- We evaluate our designed M3D DDR4 DRAM architectures using Gem5 [18] based full-system simulations with PARSEC benchmarks [17], and compare their performance and energy-efficiency with the conventional 2D DDR4 DRAM.

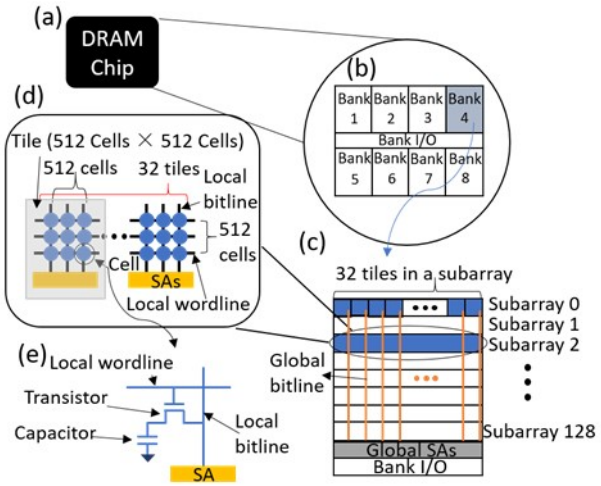


Figure 2.1: Schematic structures of (a), (b) a DRAM chip, (c) a DRAM bank, (d) a DRAM subarray, and (e) DRAM cell. SAs: Sense Amplifiers.

2.2 Background on DRAM Structure and Operation

2.2.1 DRAM Chip Structure, Operation, and Timing Constraints

A DRAM chip typically employs a hierarchical cell-array organization, which is briefly illustrated in Fig. 1. A cell is the smallest unit in the hierarchy, and the critical path for accessing a cell includes a local bitline, a local SA, a global bitline, a global SA, and bank I/O (Figure 2.1). Figure 2.2 illustrates three DRAM operation phases (activation, data I/O, and precharging), along with their related DRAM timing parameters and activities during these phases that occur in various DRAM structures, such as DRAM array, peripherals, command bus, and data bus. The definitions of various DRAM timing parameters and the DRAM structures that dominate the latency contributions for respective timing parameters are listed in Table 2.1.*From Table 2.1, lengths of local and global bitlines are major contributors to all critical access latency parameters.*

2.2.2 Latency-Area Tradeoffs for 2D DRAMs

From Table 2.1, having shorter local bitlines is the funda-mental approach for reducing tRCD, tCAS, tRP, and close-page access latency (tRCD+tCAS+tBURST). However, from [53], reducing the length of local bitlines comes at the cost of exacerbated

Table 2.1: Various DRAM Timing Parameters

Timing Parameters	Descriptions	DRAM Structure that Mainly Contributes to the Delay
tRCD	Row to Column Command Delay	Local Bitline
tCAS	Column Access Strobe Latency	Global Bitline, I/O
tBURST	Data Burst Duration	Interface
tRAS	Row Access Strobe	Local and Global Bitline, I/O, Interface
tRP	Precharge Delay	Local Bitline

latency-area tradeoffs. To evaluate these latency-area tradeoffs for different local bitline lengths, we evalu-ated the die area, tRCD and close-page access latencies for iso-capacity DDR4 [7] bank organizations with 512, 256, 128, 64, 32 cells per local bitline (respectively referred to as DDR4-512, DDR4-256, DDR4-128, DDR4-64, and DDR4-32), using our CACTI [14] and SPICE [13] based DDR4 models discussed in Section IV. The results of our evaluation are plotted in Figure 2.3. Figure 2.3 also plots results for our proposed M3D organi-zations, which will be discussed in the next section. From Figure 2.3, as we move from DDR4-512 to DDR4-32, tRCD does not reduce beyond DDR4-128 without drastic ($>2\times$) increase in die area. This is because, as move from DDR4-512 to DDR4-32, a greater number of subarrays and SA stripes are required in a DRAM bank of unchanged capacity due to the shortened local bitlines, which increases the total DRAM die area. After DDR4-128, the reduction in tRCD due to the reduction in local bitline length becomes negligible, but the increase in DRAM die are still remains significant. On the other hand, as we move from DDR4-512 to DDR4-32, the close-page access latency starts increasing significantly from DDR4-128. This is be-cause, due to the increasing number of required subarrays, the length of global bitlines increases, contributing more significantly to the tCAS and close page access latencies. Thus, contrary to the observation made for tRCD, *shorter local bitlines yield longer close-page access latencies, which can result in longer average memory access latency.*

It is clear from these findings that shortening local bitlines does not help unless the global bitlines can also be shortened in concurrence, without incurring any extra die area cost. Intu-itively, global bitlines can be shortened reducing the bank size and increasing the bank count per DRAM die. However, doing so cannot come without significant decrease in the per-die cell density of DRAM. Therefore, to address this shortcoming, we take a promising alternative approach of reorganizing DRAM banks using the coarse-grained monolithic 3D integration (M3D) technology, as discussed next.

2.3 Reoganizing DRAMs with M3D Integration

2.3.1 Monolithic 3D Integration Technology

M3D technology enables sequential processing and integration of multiple tiers (mostly up to two tiers) of logic circuits on the same die. To vertically connect various components located on different M3D tiers, the M3D-integrated chips utilize monolithic inter-tier vias (MIVs) that are several orders of magnitude smaller in physical dimensions (50nm×100nm) than TSVs (1-3m×10-30m) [61]. Moreover, an MIV has 10Ω resistance and 0.2fF capacitance. This enables vertical routing of connections using MIVs with nanoscale contact pitch and negligible overheads of parasitic loading. More details on the M3D integration technology can be found in [65]. The disadvantage of M3D integration is that, due to the sequential integration process, the resistance of the required tungsten interconnects on the bottom M3D tier can increase by up to 2×, and the transistor performance on the second/top tier can degrade by 10 20% [65]. We mitigate this tier degradation issue by employing an established workaround from [65] to make the best use of the M3D technology for designing better performing DRAM organizations.

2.3.2 Design of Monolithic 3D(M3D)DRAMs

We reorganize DDR4 DRAM [7] with M3D technology. In our designed M3D DDR4 variants, to avoid performance deg-radation on M3D tiers, we place the SAs and other peripherals (e.g., write drivers, precharge units, SA I/O, local wordline drivers, address decoders) on the bottom tier, and the DRAM cell-arrays (including the DRAM interconnects such as bitlines and wordlines) on the top tier. Figure 2.4 shows schematics of DDR4-512, M3D-512 and M3D-128 organizations. Moreover, we evaluated tRCD, close-page access latency, and die area for these and other M3D organizations (M3D-512 to M3D-32) to derive the latency-area tradeoffs for M3D designs, shown in Figure 2.3. For M3D-512 (Figure 2.4(b)), placing SAs and peripherals underneath the DRAM tiles shortens global bitline length LGBL per subarray by 234F (117F for SAs + 90F for precharge units + 27F for write drivers), yielding total LGBL to be 132,969F for M3D-512, compared to LGBL of 162,687F for DDR4-512 (Figure 2.4(a)). As a result of reduced LGBL, M3D-512 achieves reduced tCAS of 8.9ns, compared to tCAS of 10.3ns for DDR4-512. Moreover, we evaluate that the area of a 128Mb M3D-512 bank is 3.2mm2, which is significantly less than the 3.9mm2 area of a 128Mb DDR4-512 bank. Along the same lines, M3D-128 reaches the pinnacle of the benefits of M3D integration (Figure 2.4(c)), for which LGBL of 142,569F and LLBL of 256F are achieved (Figure 2.4(c)). These values of LGBL and LLBL are 1.14× and 4× less respectively than the LGBL and LLBL values for DDR4-512. Moreover, we evaluate that the area of a 128Mb M3D-128 bank is 3.4mm2, which is only 0.2mm2 less than the 3.2mm2 area of a 128Mb M3D-512 bank. Due to these benefits, the tRCD and close-page access latency curves for the M3D organizations are closer to the origin than the curves for the DDR4 organizations (Figure 2.3), which indicates that the M3D organizations relax the fundamental latency-area tradeoffs for DRAM design. *These results corroborate the excellent capabilities of the*

M3D technology in mitigating the fundamental latency-area tradeoffs for DRAMs, to achieve simultaneous benefits in DRAM access latency and per-die cell density. **Implementation Overheads for M3D DRAM Organizations:**To implement our proposed M3D DRAM organizations, we route the connections of the SAs and other peripherals on the bottom tier to the DRAM interconnects on the top tier using MIVs and tier-specific metal-via stack. Figure 2.5 illustrates the MIV-based vertical interconnects' cross-sections. Evidently, each vertical connection includes one M1-M5 metal-via stack and an MIV. We extract the parasitic resistance and capacitance values for the vertical interconnects from [56] to be 0.23fF and 20 for the worst-case scenario (i.e., highest parasitic loading) shown in Figure 2.5(c). In addition, our M3D organizations also suffer from the performance degradation of the DRAM cell access transistors placed on the top tier. We evaluate this degradation in terms of ION-IOFF characteristics using the methods from [65]. We incorporate the vertical intercon-nects' parasitic values and the degraded access transistors' ION-IOFF characteristics in our LTSpice model from [20], to evaluate their impact on various DRAM latency parameters such as tRCD and tRP. Figure 2.6(a) shows the results of our LTSpice simulations for tRCD parameter extraction for the DDR4-512, M3D-512, and M3D-128 organizations. As discussed earlier, both DDR4-512 and M3D-512 have the same value of 1024F for LLBL. From Figure 2.6(a), even with the addition of parasitic overheads of vertical interconnects and performance degradation of the access transistor, tRCD latency for M3D-512 hardly changes significantly compared to the tRCD latency for DDR4-512.*From these findings, we can conclude that M3D integration incurs negligible overhead for our proposed M3D DRAM organizations.*

2.4 Area, Timing, and Energy Analysis

We modeled various DRAM organizations for 22nm technology node using CACTI [14]. Each DRAM cell consumes $6F^2$ area, while the height and pitch of a SA are 117F and 6F respectively. We evaluate the lengths of local and global bitlines also using CACTI based models of various DDR4 and M3D organizations. For M3D organizations, we hide the area consumed by the SAs and other peripherals, to come up with bank and DRAM die area. We extract energy values from CACTI based models as well. Moreover, to evaluate various DRAM latency parameters and close-page access latency, we use the sense amplifier with DRAM subarray bitline model from [20] in LTspice [13]. The model from [20] is for 45nm, so we scale it to 22nm following the standard scaling guidelines for wires and interconnects in CMOS technologies. Our extracted modeling parameters are listed in Table 2.2 for various DDR4 and M3D DRAMs.

2.5 Simulation Setup and Results

We performed trace-driven simulations using NVmain [66] to compare the power and energy-delay product values for our considered DRAM organizations. We consider the iso-area organizations DDR4-512, M3D-512, and M3D-128 for system-level comparison. We also perform full-system simulations in Gem5 [18], to evaluate cycles per

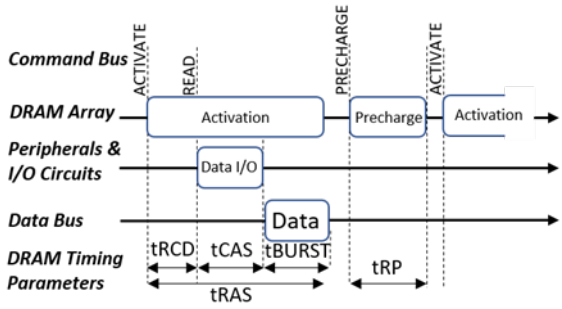


Figure 2.2: Three phases of DRAM operation and related timing parameters.

Table 2.2: MODELING PARAMETERS FOR VARIOUS DDR4 AND M3D DRAM ORGANIZATIONS.

	DDR4-512	DDR4-2T-512	DDR4-2T-128
Ranks	1	1	1
Banks	8	8	8
Page Size	16kb	16kb	16kb
Cells per Bitline	512	512	128
Timing Parameters(ns)			
tRCD	6.77	6.78	4.2
tCAS	10.29	8.96	9.82
tRP	9.58	9.6	4.04
tRC	26.64	25.34	18.05
tFAW	35.8	35.3	14.4
tREFI	7800	7800	7800
Per Access Energy Values(nJ)			
Activation Energy	0.59	0.58	0.24
Read Energy	1.1	0.94	1.05
Write Energy	1.1	0.94	1.05
Refresh Energy	35.22	32.51	23.23
Area Analysis			
Subarray(mm2)	0.031	0.025	0.007
Bank (mm2)	3.926	3.209	3.42
#MIVs per Bank	0	5,243,008	14,680,576
MIV Area Per Bank(mm2)	0	0.01	0.029
Subarray Height	1281F	1047F	279F
Local Bitline Length	1024F	1024F	256F
Local Bitline Resistance	20000Ω	20010Ω	5010Ω
Local Bitline Capacitance	72fF	72.2fF	18.2fF

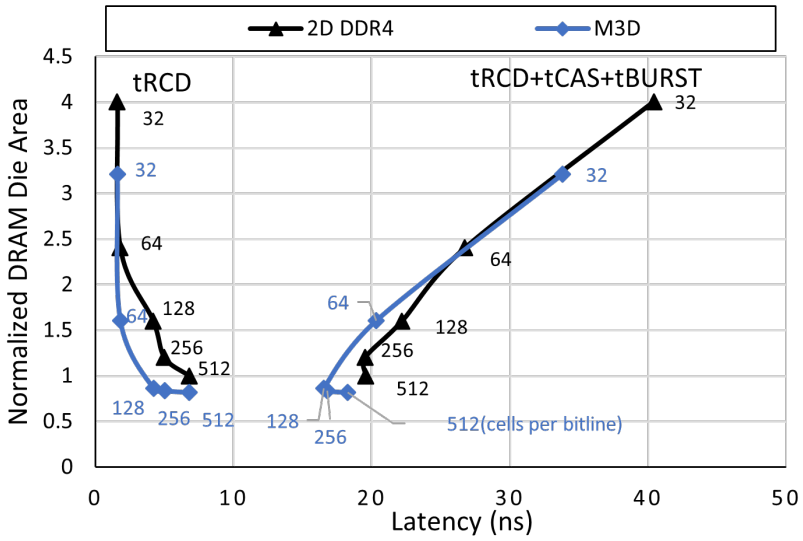


Figure 2.3: Normalized DRAM die area versus tRCD and close-page access la-tency (tRCD + tCAS + tBURST) for various local bitline lengths (cells per local bitline) for conventional 2D and M3D-enhanced DDR4 DRAM.

Table 2.3: GEM5 Configuration for Trace-Driven and Full-System Simulations.

Number of Cores	4	L2 Coherence	MOESI
L1 I Cache	32	Frequency	2 GHz
L1 D Cache	32KB	Issue policy of cores	OoO(4 issue)
Shared L2 Cache	2MB	#Memeory Controllers	1
ISA/OS	ALPHA	Cache Associativity	4-way(L1); 8-way(L2)

instruction (CPI) and average latency results. We used the PARSEC benchmarks [10] for the analysis, the trace files were extracted from detailed cycle-accurate simulations using GEM5 [18]. The configuration of GEM5 for both trace-driven and full-system simulations is shown in Table 2.3. We considered 10 different applications form the PARSEC suite: Blackscholes, Bodytrack, Canneal, Dedup, Facesim, Ferret, Stream-cluster, Swaptions, Vips, and X264. For the trace-driven simulations, we ran each PARSEC benchmark for a "warm up" period of one billion instructions and captured memory access traces form the subsequent one billion instructions extracted. For the full-system simulations, we run PARSEC benchmarks in their critical regions of interest (ROIs) in Gem5. We use parameters from Table 2.2 to model the DDR4-512, M3D-512, and M3D-128 organizations in Gem5 and NVMain. Figure 2.7(a) shows system-level cycle per instruction (CPI) values for our consid-ered DRAM organizations across PARSEC benchmarks. Compared to the baseline DDR4-512, M3D-512 and M3D-128 organizations yield about 0.54% and 3.74% lower system CPI respectively. Similarly, Figure 2.6(b) shows average access latency values. Compared to the baseline DDR4-512, M3D-512 and M3D-128 organizations yield about 1.65% and 9.56% less average latency respectively. Shorter tRC time and shorter close-page ac-

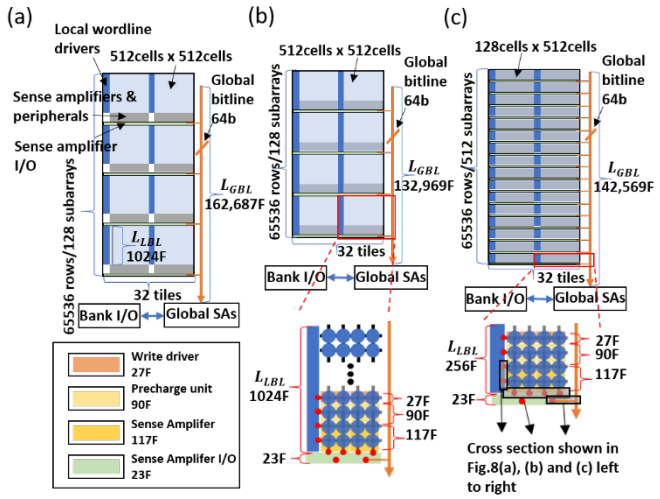


Figure 2.4: Illustration of three example bank organizations of the folded-bitline DRAM; (a) 512 cells per local bitline 2D DDR4 DRAM (DDR4-512), (b) 512 cells per local bitline M3D DRAM (M3D-512), and (c) 128 cells per local bitline M3D DRAM (M3D-128). Although the local/global address decoders are not shown, they are placed on the bottom tier.

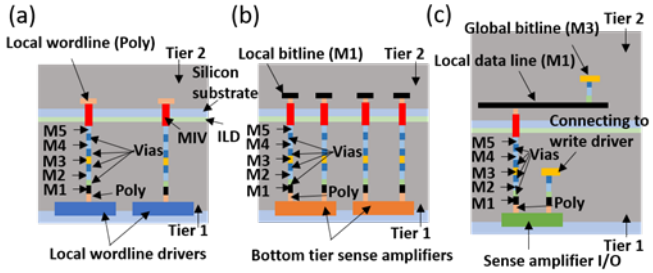


Figure 2.5: Illustration of the vertical interconnects' cross-sections between (a) local wordline drivers and local wordlines, (b) sense amplifiers (SAs) and local bitlines, and (c) sense amplifier (SA) I/O and local data line global bitline, for our proposed M3D DRAMs. ILD: Inter Layer Die electric; MIV: Monolithic Inter-tier Vias. Although the lo-cal/global address decoders are not shown, they are placed on the bottom tier.

cess latencies for the M3D-512 and M3D-128 organizations result in lower CPI and average latency values for them. Figure 2.7(b) shows energy-delay product (EDP) values. EDP indicates how balanced different designs are in terms of energy consumption and delay. We calculate EDP by multi-plying energy per bit (pJ/bit) with average access latency (ns), while energy per bit is total power divided by throughput (bit/s). The results show that M3D-512 and M3D-128 respec-tively have 7.49% and 21.21% lower EDP than the baseline DDR4-512.

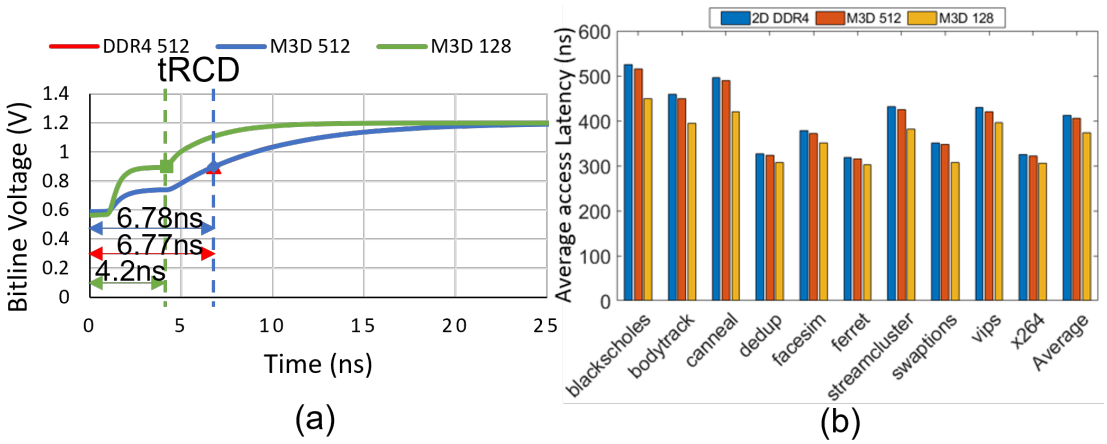


Figure 2.6: (a) Results of LTspice simulations for tRCD extraction for the DDR4-512, M3D-512 and M3D-128 organizations, and (b) average access latency results for the DDR4-512 (blue), M3D-512 (red), and M3D-128 (yellow) organizations across PARSEC benchmarks.

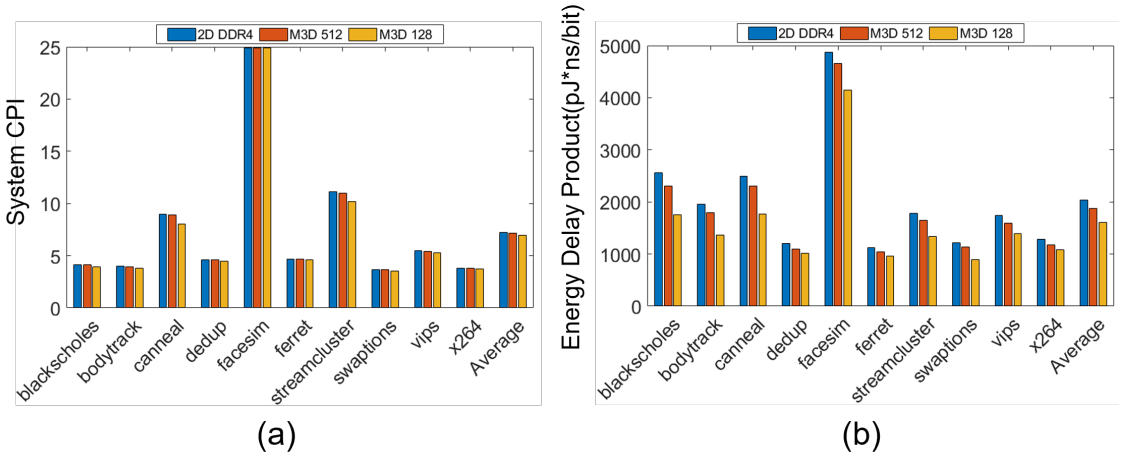


Figure 2.7: (a) System cycles per instruction (CPI), and (b) energy-delay product (EDP) results for the DDR4-512 (blue), M3D-512 (red), and M3D-128 (yellow) organizations across PARSEC benchmarks.

2.6 Conclusions

In this chapter, we showed how the fundamental latency-area tradeoffs for DRAM can be mitigated by reorganizing DRAM cell-arrays using the emerging monolithic 3D (M3D) integration technology. We evaluated the latency-area tradeoffs for various configurations of 2D DDR4 and M3D DRAMs. Based on our evaluation results for PARSEC benchmarks, we found that our designed M3D DRAM cell-array organizations can yield up to 9.56% less latency and up to 21.21% less energy-delay product

(EDP), with up to 14% less DRAM die area, compared to the conventional 2D DDR4 DRAM. These results corroborate the excellent capabilities of the M3D technology in mitigating the fundamental latency-area tradeoffs for DRAMs, to achieve simultaneous benefits in DRAM access latency and per-die cell density.

Chapter 3 Mitigating Write Disturbance in Phase Change Memory Architectures

In this chapter, we discuss our second contribution that focuses on mitigating write disturbance in Phase Change Memory (PCM) architectures to consequently enable the PCM technology, which offers inherently improved (more relaxed) density-latency trade-offs compared to DRAM, as a viable DRAM replacement for implementing main memory subsystems.

3.1 Background and Motivation

A Phase Change Memory (PCM) cell embeds a resistive heater and a small volume of chalcogenide material Ge2Sb2Te5 (GST) between two electrodes [28], as shown in Figure 3.1(a). The GST volume can be programmed into two different states (i.e., crystalline and amorphous) with dramatically different electrical resistance. The amorphous high-resistance (usually in the M range) state represents a "0", while the crystalline low-resistance (usually in the K range) state represents a "1". To write a PCM cell, two basic operations, RESET and SET, are needed. The SET operation that writes a "1" into the PCM cell requires a long-duration and low-amplitude current pulse ISET (Figure 3.1(c)). On the other hand, the RESET operation that writes a "0" into the PCM cell requires a short-duration and high-amplitude current pulse IRESET (Figure 3.1(c)), to program the GST volume into the amorphous state. While a SET operation does not typically cause any disturbance, a reliable RESET operation can heat up the GST volume to 1226.85°C [38] locally, which can dissipate excessive amount of heat into the neighboring PCM cells. *This dissipated heat can cause write disturbance (WD) errors in the neighboring PCM cells, which can potentially change their information content.*

3.1.1 Write Disturbance (WD) in PCM Cell-Array

Figure 3.1(b) shows a PCM cell-array, where PCM cells are arranged in multiple wordlines (WLs) and bitlines (BLs). The "aggressor", shown in Figure 3.1(b) at the center, is a PCM cell that is undergoing a RESET operation. The excessive heat generated by this aggressor cell dissipates into the neighboring PCM cells, in the same wordline as well as in the adjacent wordlines, increasing their temperature. If these affected neighboring cells are in the RESET state (storing "0"s), their amorphous GST volumes can become partially crystalline due to the increase in their temperature, making them the "victims" of the aggressor cell. The "0"s stored in these victim cells can be erroneously read as "1"s, if the partial crystallization of their GST volumes significantly reduces their resistance. The probability (F) of such readout error, which is referred to as write disturbance (WD) error henceforth, to occur in a victim cell is given by (3.1) [38].

$$F = 1 - exp(-(\frac{t_{fail}}{t_0})^{\beta}) \tag{3.1}$$

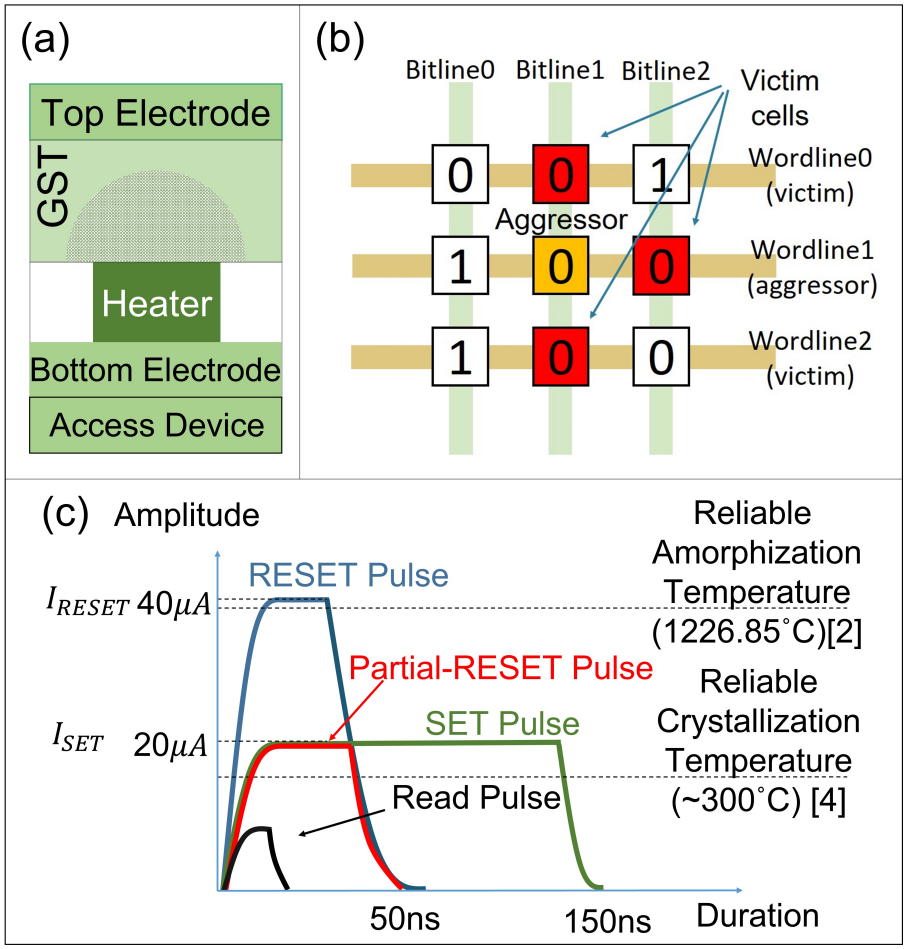


Figure 3.1: (a) Basic structure of a PCM cell; (b) A schematic of PCM cell-array; (c) PCM programming pulses from [87].

3.1.2 Related Work

To mitigate WD, conventional designs (e.g., DIN [38], SD-PCM [90], and ADAM [84]) focus on reducing the number of aggressor and victim cells per PCM write of a cacheline, and also employ Verify and Restore (VnR) mechanism to iteratively rewrite the affected cachelines, until all WD errors are recovered. DIN and ADAM employ frequent data compression to reduce number of "0"s in the compressed data, to ultimately reduce the number of aggressor and victim cells. SD-PCM uses data compression to reduce victim cells, and additionally, it employs error-correcting codes to recover from inflicted WD errors. The VnR mechanism used in these techniques incurs very high energy and latency overhead. To mitigate this overhead, ADAM retrieves the 'victim' cachelines from the last level cache, relaxing the need of using VnR [84].

3.2 Partial-Reset

To mitigate WD, we propose to use partial-RESET operations instead of full-RESET operations to write "0"s. Unlike prior works that focus on reducing the aggressor and victim cells per PCM write operation, partial-RESET focuses on eliminating the root cause of WD, that is, the excessive heat dissipation from the aggressor cells. A partial-RESET operation uses a lower-amplitude short-duration current pulse (red curve in Figure 3.1(b)) to write "0"s, instead of the high-amplitude RESET pulse. As shown in Figure 3.2, a partial-RESET pulse programs a PCM cell's GST volume in a poly-crystalline state, which renders lower resistance than the amorphous state (the full-RESET state). Nevertheless, the poly-crystalline PCM state (partial-RESET cell) still renders 10-20× higher resistance than the crystalline PCM state (SET cell), which can provide enough readout margin (Figure 3.2) to distinguish the PCM cells storing "1"s (SET cells) from PCM cells storing "0"s (partial-RESET cells). Moreover, even if the resistance of a partial-RESET cell drifts (increases) with time due to the atomic restructuring of the GST volume [87] (Figure 3.2), the readout margin remains unviolated (Figure 3.2), allowing the partial-RESET cell to be unerringly distinguished from the SET cell. We think that the polycrystalline partial-RESET state of a PCM cell (Figure 3.2) is deterministically achievable with reasonable repeatability, as it is analogous to an intermediate resistance state of a multi-level (MLC) PCM cell [81], which is traditionally achieved with deterministic repeatability by applying a current pulse with intermediate amplitude and/or width [19].

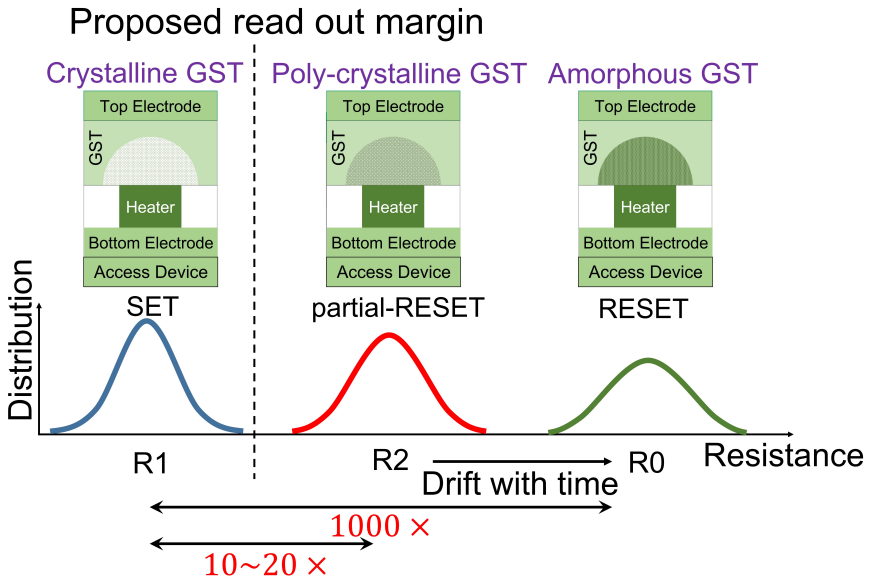


Figure 3.2: Illustration of partial-RESET, SET, and RESET PCM cells, and their resistance distributions.

The lower-amplitude current pulse used for the partial-RESET operation hardly increases the temperature of an aggressor cell above the crystallization point (300°C),

which results in negligible heat dissipation from the aggressor cell into the victim cells. This in turn increases the value in (3.1) to 50 [38], rendering a negligible value of 10-12 for the WD error probability F. Thus, with nearly zero WD error probability, partial-RESET operations effectively eliminate the WD problem in PCM architectures. Moreover, the use partial-RESET operations eliminates the need of traditional VnR mechanism [38] [90] [84], and as a result, saves PCM architectures from the excessive latency and energy overheads related to the VnR mechanism.

3.3 Results

For our trace-driven evaluations with PARSEC benchmarks [7], we use the PCM configuration and simulation environment from [84]. Moreover, we employ the method from [38] to evaluate per-write energy and latency values for the baseline, ADAM [84], and our proposed partial-RESET PCM architectures, for 20nm PCM technology. Figure 3.3(a) shows number of total WD errors (BL+WL WD errors) per-write for three PCM architectures across three benchmarks. As evident, partial-RESET has zero WD errors. Figure 3.3(b) and 3.3(c), respectively, present per-write energy and latency, for three PCM architectures. Our partial-RESET PCM does not require expensive VnR mechanism, which yields the least per-write energy and latency values for partial-RESET, compared the baseline and ADAM.

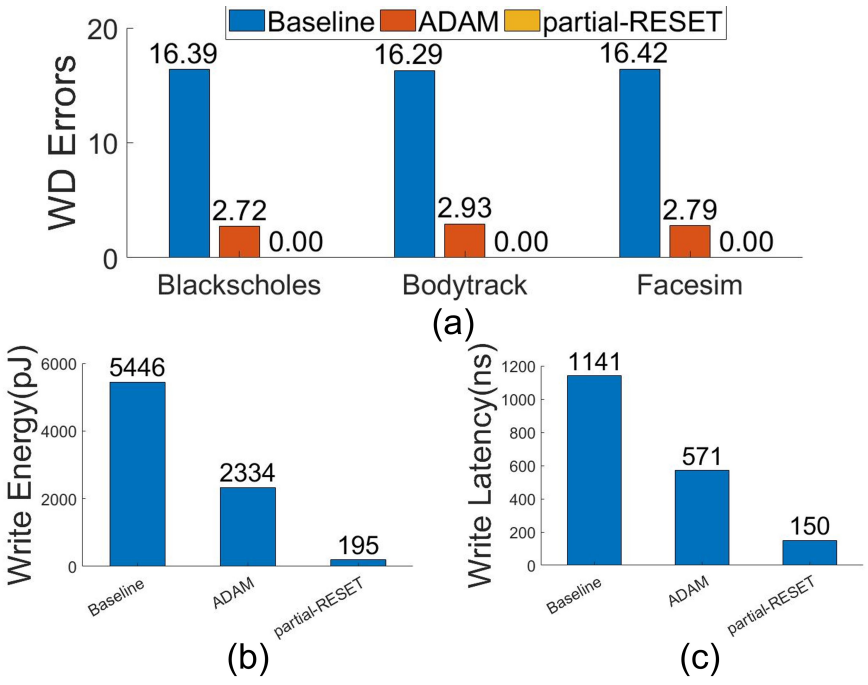


Figure 3.3: (a) Number of total WD errors (BL+WL WD errors) across three PARSEC benchmarks, (b) energy, and (c) latency, per-write for the baseline, ADAM, and partial-RESET PCM architectures. The bars for partial-RESET in (a) are not visible due to their zero heights.

3.4 Limitations of PCM Architectures with partial-RESET Operations

The major drawback of partial-RESET operations would occur when they would be used to implement multi-level cell (MLC) PCM architectures. In MLC PCM, every PCM cell can store multiple bits (typically 2 bits) of information. This is made possible by dividing the large resistance range of PCM cells (typically from a few KOhms to a few MOhms) into multiple discrete resistance bands (typically four resistance bands) that are separated by viable inter-band gaps to provide sufficient readout margins. These inter-band gaps can guard MLC PCM architectures from readout errors when the inherent drift occurring in the resistance of PCM cells [58] forces the neighboring resistance bands to overlap with one another to consequently violate the readout margins. To this end, the use of partial-RESET operations for implementing MLC PCM architectures would reduce the number of bits that can be stored per PCM cell. This is because, the use of partial-RESET operations would reduce the dynamic resistance range of PCM cells by setting the highest resistance state to be of a few hundred KOhms as opposed to a few MOhms. As a result, if the inter-band gaps are kept unchanged to ensure sufficient readout margins, the number of distinct resistance bands that can be accommodated in this reduced dynamic resistance range would also be less, resulting in a less number of bits that can be stored per PCM cell.

3.5 Conclusions and Future Work

This chapter shows that the use of partial-RESET operations can eliminate write disturbance (WD) errors, in addition to improving the energy-efficiency and latency of reliable write operations, in SLC PCM architectures. Thus, partial-RESET operations represent a promising solution for mitigating WD in the scaled SLC PCM implementations of the future. Going forward, we plan to perform a detailed system-level simulation analysis of partial-RESET operations with more number of benchmark applications. Furthermore, we intend to explore the usefulness as well as drawbacks of using partial-RESET operations for mitigating WD errors in MLC PCM architectures.

Chapter 4 Conclusion and Future Work

4.1 Conclusion

In this thesis, we provided possible solutions for improving main memory subsystems using emerging technologies, including monolithic 3D technology for DRAM latency-area tradeoff and partial-reset for PCM write disturbance. In Chapter 2, we presented how the fundamental latency-area tradeoffs for DRAM can be mitigated by reorganizing DRAM cell arrays using the emerging monolithic 3D(M3D) integration technology. Based on our evaluation results for PARSEC benchmarks, we found that our designed M3D DRAM cell-array organizations can yield up to 25.71% less latency and up to 43% less energy-delay product(EDP), with up to 14% less DRAM die area, compared to the conventional 2D DDR4 DRAM.

In Chapter 3, we first discussed the PCM as a potential candidate to replace DRAM as main memory, and then highlighted write disturbance as one of the issues that harm the PCM reliability. To solve this issue, we presented how our proposed technique Partial-RESET eliminates the root cause of write disturbance, which in turn eliminates write disturbance errors. Moreover, we showed that Partial-RESET operations induce significantly low energy and latency overheads, which can significantly improve the performance and energy-efficiency of PCMs, compared to the relevant prior works.

Both of our proposed ideas, the M3D DRAM and Partial-RESET based PCM, solve the fundamental problems of DRAM based main memory subsystems. Our results corroborate that the emerging technologies such as M3D and PCM are worth exploiting and could bring significantly better performance to main memory subsystems.

4.2 Future Research Directions

M3D Integration Based DRAM Design: An extension of our proposed 2-Tier M3D DRAM would be a 3-Tier M3D DRAM. The motivation is to extract more benefits by further shortening the bitlines without corresponding increase in DRAM area, to consequently reduce the fundamental DRAM access latency. Figure 4.1 illustrates the envisioned idea. Figure 4.1(a) is showing a schematic layout of two DRAM cell arrays (Cell array 1, Cell array 2) and corresponding peripheral circuits (SA1, SA2, SA I/O 1, SA I/O 2) for the 2-Tier M3D design with 128 cells per bitline. Figure 4.1(b) shows a schematic layout of two DRAM cell arrays (Cell array 1, Cell array 2) and corresponding peripheral circuits (SA1, SA2, SA I/O 1, SA I/O 2) for the 2-Tier M3D design with 64 cells per bitline. Figure 4.1(c) shows a schematic layout of two DRAM cell arrays (Cell array 1, Cell array 2) and corresponding peripheral circuits (SA1, SA2, SA I/O 1, SA I/O 2) for the 3-Tier M3D design with 64 cells per bitline. In these figures, SA means sense amplifier. SA I/O 1 and SA I/O 2 show regions where vertical interconnections are made between DRAM cell arrays

and SAs as well as between SAs and global I/O circuits (not shown in the figures). Cell arrays are implemented on the top M3D tier, whereas SAs and SA I/O regions are implemented on the bottom M3D tier, just like discussed in Chapter 2. A DRAM cell array together with its corresponding SA and SA I/O circuits (e.g., cell array 1, SA 1, and SA I/O 1) makes an independently operable M3D DRAM subarray. Since the SA I/O regions are implemented on the bottom tier and are utilized for routing the inter-tier interconnects vertically within their respective M3D subarrays, no other circuit is implemented right on top of them on the top M3D tier. Doing so ensures viable routing of wires between the SA I/O regions and the cell arrays on the top M3D tier.

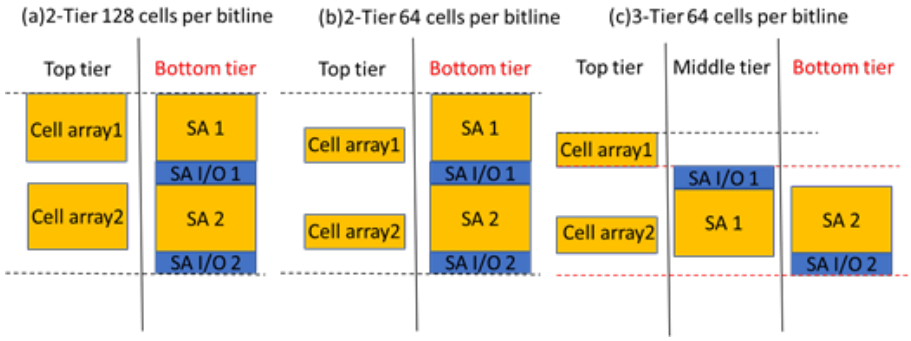


Figure 4.1: Simplified schematic illustration of (a) 2 Tier M3D DRAM design with 128 cells per bitline. (b) 2 Tier M3D DRAM design with 64 cells per bitline. (c) 3 Tier M3D DRAM design with 64 cells per bitline

In a nutshell, in an M3D DRAM bank (such as shown in Figure 4.1), to keep reducing DRAM access latency without increasing DRAM bank area, the bitline length can be reduced up to a point where the DRAM cell arrays of the individual M3D subarrays start occupying smaller area than their corresponding SAs. Beyond this point, further decreasing bitline slows down latency reduction but ramps up the increase in DRAM bank area. This happens because reducing bitline length always increases the required number of subarrays per bank (for a constant bank capacity), which in turn incurs extra area overhead of the corresponding increase in the number of SAs. But this extra area overhead can no longer be completely hidden under the area occupied by DRAM cell arrays, when individual DRAM cell arrays start occupying smaller area than their corresponding SAs. For instance, in case of the 2-Tier M3D design with 128 cells per bitline (Figure 4.1(a)), the area occupied by cell array 1 is equal to the area occupied by SA 1, and it is true for cell array 2 and SA 2 as well. Therefore, when the area occupied by cell array 1 becomes smaller than the area occupied by SA 1, in case of the 2-Tier M3D design with 64 cells per bitline (Figure 4.1(b)), the latency reduction starts slowing down and the increase in bank area starts ramping up. This ramp up in bank area can be notably slowed down by adding more M3D tiers in the DRAM design. For example, adding one more M3D tier (having total 3 tiers) in Figure 4.1(c) can reduce the area overhead of having 64 cells per bitline compared to the 2-Tier design shown in Figure 4.1(b), by splitting

the placement of SAs across the bottom two M3D tiers. This way, the 3-Tier M3D design can continue to provide viable latency and area benefits for bitlines that are shorter than 128 cells. Therefore, 3-Tier M3D concept is worth exploring.

However, the M3D integration technology intrinsic challenges may get a bit more amplified in the 3-Tier M3D design compared to the 2-Tier M3D design. This is because the tungsten based interconnects on the middle and bottom tiers of the 3-Tier M3D DRAM would suffer from high resistivity. In addition, the transistors implemented on the top tier and middle tier would suffer from the degraded performance. Thus, the middle tier may suffer from degraded interconnects as well as transistor performance. Nevertheless, if the performance degradation can be kept below the achievable latency benefits, the 3-Tier M3D DRAM design is worth exploring. In fact, as the M3D integration technology improves in the future, it might be possible to include more than three tiers per chip for DRAM design. Doing so may provide even more latency and area benefits in the future. Moreover, there is one more opportunity for further improvement in M3D DRAM implementations. As the M3D integration technology improves, if we are able to reduce the performance degradation of tungsten based interconnects and deposited transistors, there won't be any doubt that the M3D integration based DRAM architectures will proliferate in the future.

Partial-RESET Based PCM: The future research direction for the Partial-RESET PCM design will likely be the multi-bit cell design. The ability to store multiple bits per PCM cell is part of the reason for the PCM technology being a candidate for replacing DRAM as main memory. A multibit cell PCM's ability of increasing the bit density without changing the cell density, and hence without correspondingly increasing the latency overhead, provides a tempting solution for main memory implementation. Carefully optimizing the Partial-RESET could lead the way to achieve multibit cell PCMs without the write disturbance issue. In addition, the elimination of the write disturbance issue would allow PCM cells to be placed closer to each other by decreasing the wordline and bitline pitches. This could lead to more compact PCM, multiplying the latency and density benefits of PCM over DRAM.

Appendices

Appendix A: LTSpice Based Modeling of M3D DRAMs' Bitline-Level Organizations

Introduction

To evaluate how the bitline length affects various DRAM timing constraints and close-page DRAM access latency, we chose to model DRAM cell-arrays using LTSpice following the guidelines from [20]. We also use LTSpice to evaluate how the degradation of transistors and interconnects performance in M3D technology affect various DRAM timing parameters and DRAM access latency.

Modeling DRAM Cell Array

As discussed in Chapter 2, a DRAM cell array basically consists of a 2D array of 1-transistor-1-capacitor (1T1C) type of DRAM cells that are connected in the 2D array through bitlines (vertical wires) and wordlines (horizontal wires). Each individual DRAM cell capacitor connects to a bitline through an access transistor that can be switched ON/OFF by enabling/disabling a wordline. Figure 1 shows the full view of the DRAM cell-array organization of DDR4 DRAM that we have extracted using the LTSpice simulation based model from [20]. Different parts of this cell-array (i.e., bitlines, sense-amplifiers, access transistors) are remodeled as follows to fit the requirements of our designed M3D DRAMs.

DRAM cell access transistors: The original model of the access transistors are taken from [20] which provides a 44 nm node model for DDR4 DRAM. We scale this model to 22 nm for our M3D DRAM design using guidelines from cmosptm [34]. The original V_{DD} of the model is 1.5V, whereas for 22 nm we reduce V_{DD} to 1.2V. To model the degradation of the transistors that are placed on the top M3D tier of our M3D DRAM design, we propose lowering the I_{ON} of the transistors by 20% according to [65].

Sense-amplifiers and other peripherals: Sense-amplifiers and other peripherals contain cross-coupled inverters as sense-amplifiers and other circuit blocks that act as precharge units. Figure 2 shows a sense-amplifier and the corresponding precharge unit. The transistor models utilized for simulating these the sense-amplifier and precharge unit blocks are also taken from [34] and [20] for 22 nm node. Nevertheless, in contrast to the DRAM cell access transistors, the sense-amplifers and precharge units do not face any degradation as they are implemented on the bottom M3D tier.

Bitlines and wordlines: At the very top of the Figure 1, we have bitline and wordline of DRAM array 0 modeled as lumped RC parameters. At the bottom right corner, we have bitline and wordline of DRAm array 1. Focusing on array 0, R1 represents the bitline resistance and Cmbit2 represents the bitline parasitic capacitance. Similarly, R2 and Cmbit3 are the bitline resistance and capacitance

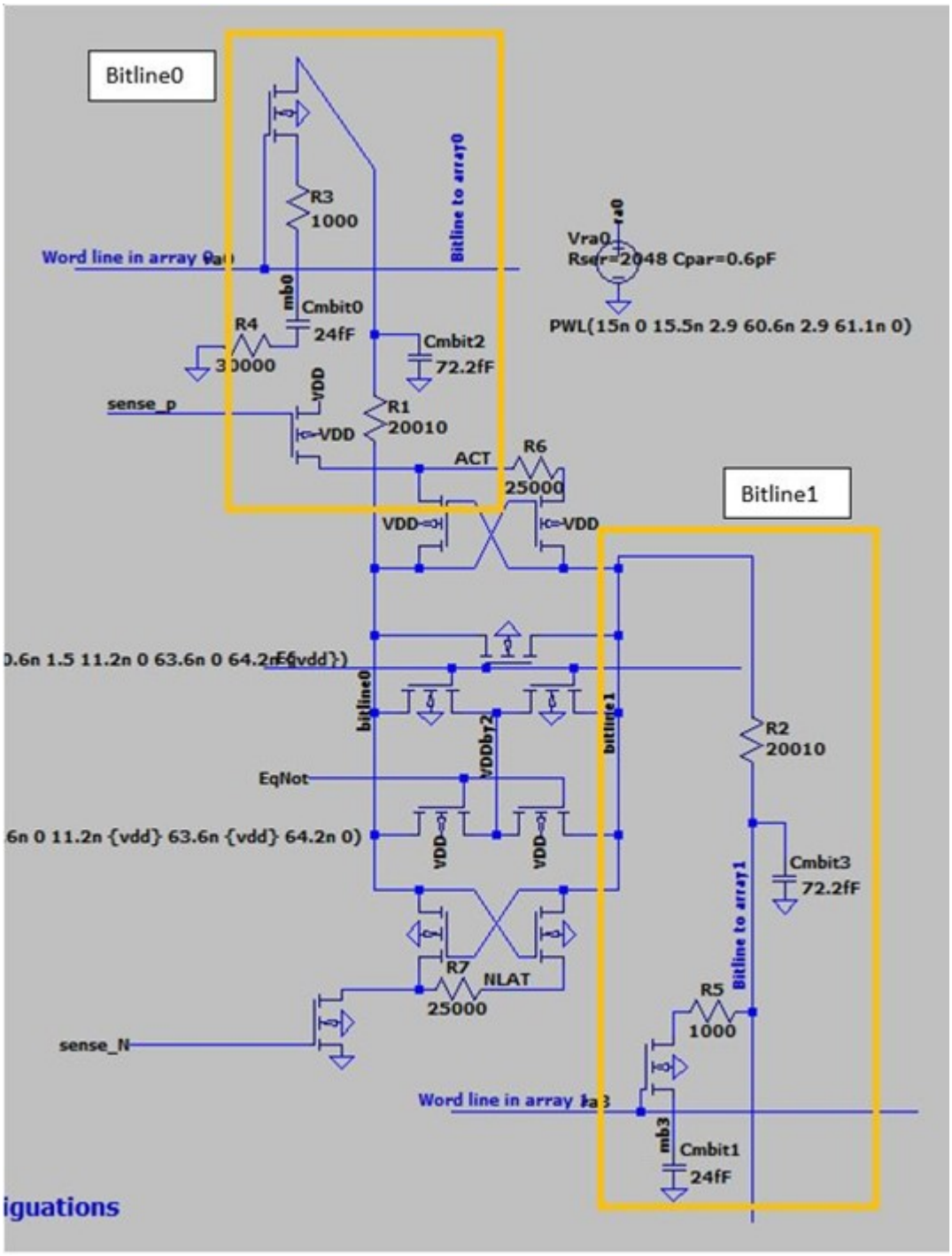
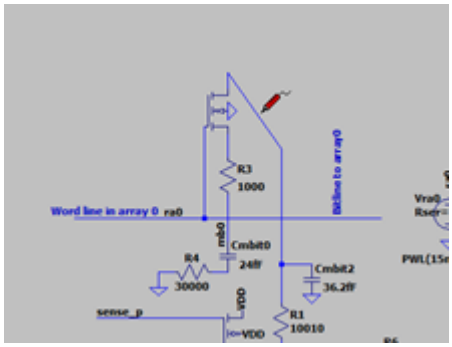


Figure 1: The LTspice model for DRAM bitline and sense amplifier. Bitline0 and bitline1 marked in yellow.

values for array 1. To simulate various number of cells per bitline, we have the parameters R1, R2, Cmbit2, Cmbit3 set according to Table 1.

Simulation Results Steps

- All the parameters that are shown in Figure 1 and 2 can be set by clicking on their values and modifying them.
- After all parameters are set. Click "run" to start to simulation.
- To check the bitline voltage, click on the bitline to see the voltage waveform. As you can see by the red pin shown in the figure below.



- Click twice on the name of the waveform to show two cursors, move the cursors to show the details of the waveform.
- For tRCD, we collect the data from half V_{DD} to 0.75 V_{DD}(0.9V).
- For tRP, we collect the data from full V_{DD} to 0.51 V_{DD}(0.612V).
- The box in the right bottom will show the data set you selected using the cursors.

Data Collection

The goal of the LTspice is to simulate the timing parameters of DDR4 and M3D DRAM designs. Two timing parameters, tRCD and tRP, are extracted from LTSpice. We use the method from [20] to extract tRCD and tRP. According to [20], tRCD is the interval for which the voltage of the bitline is charged up to 75% of V_{DD} (0.9V) after a read command is issued. And tRP is the time interval for which the voltage of the bitline is reduced/discharged t0 51% of V_{DD}(0.612V) after a precharge command is issued.

Table 1: Parameters for different bitline organizations. R1/R2 are bitline resistance values in Ohms and C1 (Cmbit2) and C2 (Cmbit3) are bitline capacitance values in fF. Parameters R1, R2, C1 (Cmbit2), C2 (Cmbit3) are shown in Figure 1.

Cells per bitline	DDR4		2T-M3D		3T-M3D	
	R1/R2	C1/C2	R1/R2	C1/C2	R1/R2	C1/C2
512	20000	72	20010	72.2	20020	72.4
256	10000	36	10010	36.2	10020	36.4
128	5000	18	5010	18.2	5020	18.4
64	2500	9	2510	9.2	2520	9.4
32	1250	4.5	1260	4.7	1270	4.9

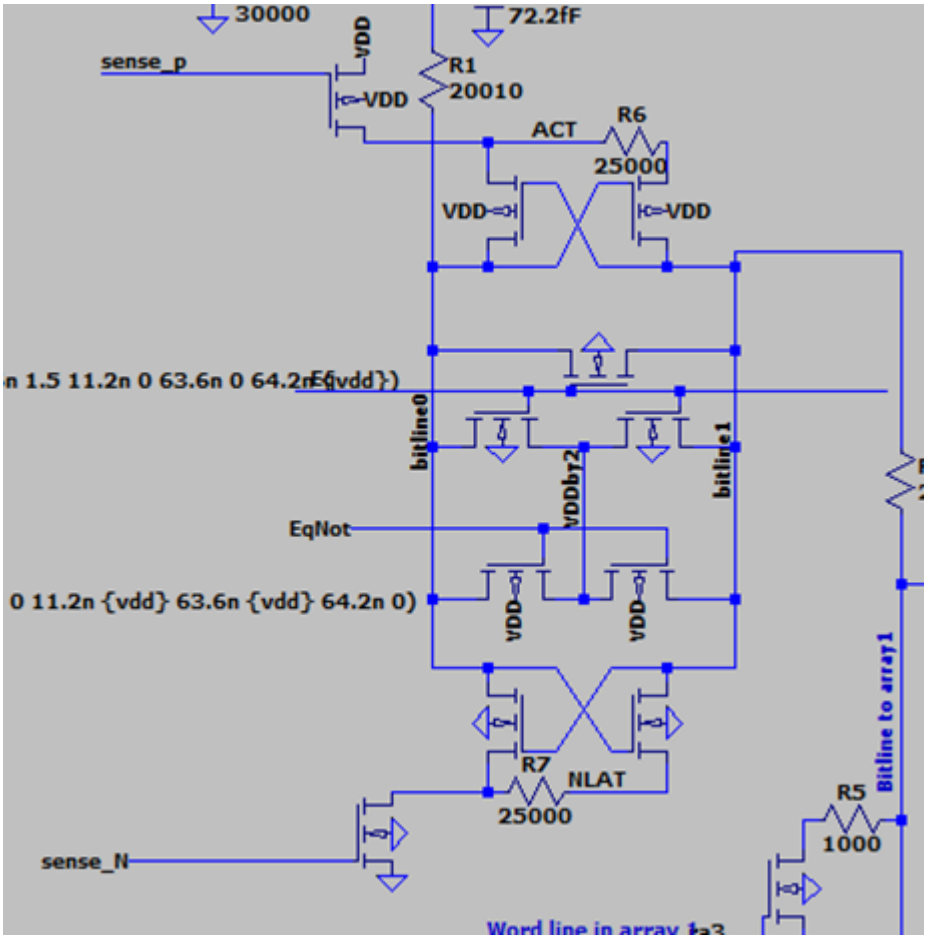


Figure 2: LTspice model for sense amplifier and peripherals

Bibliography

[1] *Enhanced Memory Systems Enhanced SDRAM SM2604*. 2002.

[2] Nec.virtual channel sdram upd4565421. 1999.

[3] Process integration, devices and structures. *International Technology Roadmap for Semiconductors*, 2013.

[4] H. Aghaei Khouzani, F. S. Hosseini, and C. Yang. Segment and conflict aware page allocation and migration in dram-pcm hybrid main memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9):1458–1470, 2017.

[5] J. H. Ahn et al. *Improving system energy eXciency with memory rank subsetting*. ACM TACO, 2012.

[6] J. H. Ahn, J. Leverich, R. Schreiber, and N. P. Jouppi. Multicore dimm: an energy efficient memory module with independently controlled drams. *IEEE Computer Architecture Letters*, 8(1):5–8, 2009.

[7] S. Akram, J. B. Sartor, K. S. McKinley, and L. Eeckhout. *Write-rationing garbage collection for hybrid memories*. Programming Language Design and Implementation (PLDI), 2018.

[8] A. E. al. Writing without disturb on phase change memories by integrating coding and layout design. In *Proceedings of the 2015 International Symposium on Memory Systems, ser. MEMSYS'15. , NY, USA: ACM*, pages 71–77, 2015.

[9] R. W. al. Exploit imbalanced cell writes to mitigate write disturbance in dense phase change memory. *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2015.

[10] R. W. al. Decongest: Accelerating super-dense pcm under write disturbance by hot page remapping. *IEEE Computer Architecture Letters*, 16(2):107–110, July 2017.

[11] S. S. al. Use ecp, not ecc, for hard failures in resistive memories. pages 141–152. Proceedings of the 37th Annual International Symposium on Computer Architecture, ser. ISCA '10. , NY, USA: ACM, 2010.

[12] M. Arjomand, M. T. Kandemir, A. Sivasubramaniam, and C. R. Das. Boosting access parallelism to pcm-based main memory. In *International Symposium on Computer Architecture (ISCA)*, 2016.

[13] R. Ausavarungnirun et al. *Staged memory scheduling: achieving high performance and scalability in heterogeneous systems*. In ISCA, 2012.

[14] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas. Cacti 7: New tools for interconnect exploration in innovative off-chip memories. *ACM Trans. Archit. Code Optim.*, 14(2), June 2017.

[15] G. J. Barkley, D. Vimercati, and P. Garofalo. Apparatus and methods to perform read-while write (rww) operations. *US Patent App. /688,667*, 15, 2017.

[16] P. Batude, T. Ernst, J. Arcamone, G. Arndt, P. Coudrain, and P.-E. Gaillardon. 3-d sequential integration: A key enabling technology for heterogeneous co-integration of new function with cmos. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(4):714–722, 2012.

[17] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The parsec benchmark suite: Characterization and architectural implications. In *2008 International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 72–81, 2008.

[18] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.

[19] G. W. Burr, M. J. Brightsky, A. Sebastian, H.-Y. Cheng, J.-Y. Wu, S. Kim, N. E. Sosa, N. Papandreou, H.-L. Lung, H. Pozidis, E. Eleftheriou, and C. H. Lam. Recent progress in phase-change memory technology. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(2):146–162, 2016.

[20] K. K. Chang, A. G. Yağlıkçı, S. Ghose, A. Agrawal, N. Chatterjee, A. Kashyap, D. Lee, M. O'Connor, H. Hassan, and O. Mutlu. Understanding reduced-voltage operation in modern dram devices: Experimental characterization, analysis, and mechanisms. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1), June 2017.

[21] S. Cho and H. Lee. Flip-n-write: A simple deterministic technique to improve pram write performance energy and endurance. In *Proc. MICRO*, pages 347–357, 2009.

[22] S. Cho and H. Lee. Flip-n-write: a simple deterministic technique to improve pram write performance, energy and endurance. Symposium on Microarchitecture (micro), 2009.

[23] E. Ebrahimi et al. *Parallel application memory scheduling.* MICRO, 2011.

[24] K. Ghose and M. Kamble. Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation. In *Proceedings. 1999 International Symposium on Low Power Electronics and Design (Cat. No.99TH8477)*, pages 70–75, 1999.

[25] S. Ghose, T. Li, N. Hajinazar, D. S. Cali, and O. Mutlu. Demystifying complex workload-dram interactions: An experimental study. *Proc. ACM Meas. Anal. Comput. Syst.*, 3(3), dec 2019.

[26] C. Hart. Cdram in a unified memory architecture. In *Proceedings of COMPCON '94*, pages 261–266, 1994.

[27] H. Hidaka, Y. Matsuda, M. Asakura, and K. Fujishima. The cache dram architecture: a dram with an on-chip cache memory. *IEEE Micro*, 10(2):14–25, 1990.

[28] H. Horii, J. Yi, J. Park, Y. Ha, I. Baek, S. Park, Y. Hwang, S. Lee, Y. Kim, K. Lee, U.-I. Chung, and J. Moon. A novel cell technology using n-doped gesbte films for phase change ram. In *2003 Symposium on VLSI Technology. Digest of Technical Papers (IEEE Cat. No.03CH37407)*, pages 177–178, 2003.

[29] W.-C. Hsu and J. E. Smith. *Performance of cached DRAM organizations in vector supercomputers.* ISCA, 1993.

[30] J. Hu, C. J. Xue, Q. Zhuge, W.-C. Tseng, and E. H. m Sha. Write activity reduction on non-volatile main memories for embedded chip multiprocessors. *ACM Transactions on Embedded Computing*, 2013.

[31] C.-H. Huang and I. G. Thakkar. Mitigating write disturbance in phase change memory architectures: Work-in-progress. In *Proceedings of the International Conference on Compliers, Architectures and Synthesis for Embedded Systems Companion*, CASES '19, New York, NY, USA, 2019. Association for Computing Machinery.

[32] C.-H. Huang and I. G. Thakkar. Improving the latency-area tradeoffs for dram design with coarse-grained monolithic 3d (m3d) integration. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 417–420, 2020.

[33] Y. Huang, T. Liu, and C. J. Xue. *Register aloptlocation for write activity minimization on non-volatile main memory.* Asia South Pacific Design Automation Conference (ASP-DAC), 2011.

[34] N. Integration and A. Modeling (NIMO) Group. http://ptm.asu.edu/.

[35] J. Jeddeloh and B. Keeth. Hybrid memory cube new dram architecture increases density and performance. In *2012 Symposium on VLSI Technology (VLSIT)*, pages 87–88, 2012.

[36] L. Jiang, Y. Zhang, B. R. Childers, and J. Yang. Fpb: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory. In *Proc. MICRO*, pages 1–12, 2012.

[37] L. Jiang, Y. Zhang, B. R. Childers, and J. Yang. Fpb: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory. Symposium on Microarchitecture (MICRO), 2012.

[38] L. Jiang, Y. Zhang, and J. Yang. Mitigating write disturbance in super-dense phase change memories. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 216–227, 2014.

[39] L. Jiang, B. Zhao, J. Yang, and Y. Zhang. A low power and reliable charge pump design for phase change memories. In *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, pages 397–408, 2014.

[40] L. Jiang, B. Zhao, Y. Zhang, J. Yang, and B. R. Childers. Improving write operations in mlc phase change memory. In *IEEE International Symposium on High-Performance Comp Architecture*, pages 1–10, 2012.

[41] G. Kedem and R. P. Koganti. Wcdram: A fully associative integrated cached-dram with wide cache lines. *Duke*, 1997.

[42] B. Keeth et al. *DRAM Circuit Design Fundamental and High-Speed Topics.* Wiley-IEEE Press, 2007.

[43] Y. Kim, D. Han, O. Mutlu, and M. Harchol-Balter. Atlas: A scalable and high-performance scheduling algorithm for multiple memory controllers. In *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*, pages 1–12, 2010.

[44] Y. Kim, M. Papamichael, O. Mutlu, and M. Harchol-Balter. Thread cluster memory scheduling: Exploiting differences in memory access behavior. In *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 65–76, 2010.

[45] Y. Kim, V. Seshadri, D. Lee, J. Liu, and O. Mutlu. A case for exploiting subarray-level parallelism (salp) in dram. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*, pages 368–379, 2012.

[46] Y. Kim, S. Yoo, and S. Lee. Write performance improvement by hiding r drift latency in phase-change ram. In *Proc. DAC*, pages 897–906, 2012.

[47] T. Kimuta, K. Takeda, Y. Aimoto, N. Nakamura, T. Iwasaki, Y. Nakazawa, H. Toyoshima, M. Hamada, M. Togo, H. Nobusawa, and T. Tanigawa. 64 mb 6.8 ns random row access dram macro for asics. In *1999 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. First Edition (Cat. No.99CH36278)*, pages 416–417, 1999.

[48] S. Kwon, S. Yoo, S. Lee, and J. Park. Optimizing video application design for phase-change ram-based main memory. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(11):2011–2019, 2012.

[49] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. Architecting phase change memory as a scalable dram alternative. International Symposium on Computer Architecture (ISCA), 2009.

[50] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. Phase change memory architecture and the quest for scalability. *Commun. ACM*, 53(7):2010, 2010.

[51] B. C. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, and D. Burger. Phase-change technology and the future of main memory. *IEEE Micro*, 30(1):2010, 2010.

[52] C. J. Lee, V. Narasiman, E. Ebrahimi, O. Mutlu, and Y. N. Patt. *DRAM-aware last-level cache writeback: Reducing write-caused interference in memory systems*, 2010.

[53] D. Lee, Y. Kim, V. Seshadri, J. Liu, L. Subramanian, and O. Mutlu. Tiered-latency dram: A low latency and low cost dram architecture. In *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, pages 615–626, 2013.

[54] H. G. Lee, S. Baek, C. Nicopoulos, and J. Kim. An energy-and performance-aware dram cache architecture for hybrid dram/pcm main memory systems. In *Proc. ICCD*, pages 381–387, 2011.

[55] S. Lee, H. Bahn, and S. H. Noh. Clock-dwf: A write-history-aware page replacement algorithm for hybrid pcm and dram memory architectures. *IEEE Transactions on Computers*, 63(9):2187–2200, 2014.

[56] Y.-J. Lee, P. Morrow, and S. K. Lim. Ultra high density logic designs using transistor-level monolithic 3d integration. In *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 539–546, 2012.

[57] B. Li, S. Shan, Y. Hu, and X. Li. Partial-set: Write speedup of pcm main memory. In *Proc. DATE*, pages 1–4, 2014.

[58] J. Li, B. Luan, and C. Lam. Resistance drift in phase change memory. In *2012 IEEE International Reliability Physics Symposium (IRPS)*, pages 6C.1.1–6C.1.6, 2012.

[59] J. Li and K. Mohanram. Write-once-memory-code phase change memory. In *Proc. DATE*, pages 1–6, 2014.

[60] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu. Raidr: Retention-aware intelligent dram refresh. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12, 2012.

[61] S. Musavvir, A. Chatterjee, R. G. Kim, D. H. Kim, and P. P. Pande. Inter-tier process-variation-aware monolithic 3-d noc design space exploration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(3):686–699, 2020.

[62] O. Mutlu and T. Moscibroda. Stall-time fair memory access scheduling for chip multiprocessors. In *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, pages 146–160, 2007.

[63] O. Mutlu and T. Moscibroda. Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared dram systems. In *2008 International Symposium on Computer Architecture*, pages 63–74, 2008.

[64] C. Pan, M. Xie, J. Hu, Y. Chen, and C. Yang. 3m-pcm: Exploiting multiple write modes mlc phase change main memory in embedded systems. In *Proc. CODES+ISSS*, pages 1–10, 2014.

[65] S. Panth, S. K. Samal, K. Samadi, Y. Du, and S. K. Lim. Tier degradation of monolithic 3-d ics: A power performance study at different technology nodes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(8):1265–1273, 2017.

[66] M. Poremba, T. Zhang, and Y. Xie. Nvmain 2.0: A user-friendly memory simulator to model (non-)volatile memory systems. *IEEE Computer Architecture Letters*, 14(2):140–143, 2015.

[67] B. Pourshirazi, M. V. Beigi, Z. Zhu, and G. Memik. *WALL: A writeback-aware LLC management for PCM-based main memory systems.* Design Automation Test in Europe Conference Exhibition (DATE), 2018.

[68] B. Pourshirazi, M. V. Beigi, Z. Zhu, and G. Memik. Writeback-aware llc management for pcm-based main memory systems. *ACM Transactions on Design Automation of Electronic Systems*, 24(2):2019, 2019.

[69] M. K. Qureshi, M. M. Franceschini, A. Jagmohan, and L. A. Lastras. Preset: Improving performance of phase change memories by exploiting asymmetry in write times. In *Proc. ISCA*, pages 380–391, 2012.

[70] M. K. Qureshi, M. M. Franceschini, A. Jagmohan, and L. A. Lastras. Preset: Improving performance of phase change memories by exploiting asymmetry in write times. International Symposium on Computer Architecture (ISCA), 2012.

[71] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-Monta no. Improving read performance of phase change memories via write cancellation and write pausing. In *Proc. HPCA*, pages 1–11, 2010.

[72] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-Montano. Improving read performance of phase change memories via write cancellation and write pausing. *In High Performance Computer Architecture (HPCA)*, 2010.

[73] M. K. Qureshi, M. M. Franceschini, L. A. Lastras-Monta no, and J. P. Karidis. Morphable memory system: A robust architecture for exploiting multi-level phase change memories. International Symposium on Computer Architecture (ISCA), 2010.

[74] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. Scalable high performance main memory system using phase-change memory technology. International Symposium on Computer Architecture (ISCA), 2009.

[75] R. Rao et al. *Exploiting non-uniform memory access patterns through bitline segmentation.* WMPI, 2006.

[76] R. H. Sartore et al. Enhanced dram with embedded registers. *U.S. patent*, Patent number 5887272, 1999.

[77] Y. Sato, T. Suzuki, T. Aikawa, S. Fujioka, W. Fujieda, H. Kobayashi, H. Ikeda, T. Nagasawa, A. Funyu, Y. Fuji, K. Kawasaki, M. Yamazaki, and M. Taguchi. Fast cycle ram (fcram); a 20-ns random row access, pipe-lined operating dram. In *1998 Symposium on VLSI Circuits. Digest of Technical Papers (Cat. No.98CH36215)*, pages 22–25, 1998.

[78] N. H. Seong, D. H. Woo, and H.-H. S. Lee. Security refresh: Prevent malicious wear-out and increase durability for phase-change memory with dynamically randomized optaddress mapping. International Symposium on Computer Architecture (ISCA), 2010.

[79] V. Seshadri, A. Bhowmick, O. Mutlu, P. B. Gibbons, M. A. Kozuch, and T. C. Mowry. The dirty-block index. International Symposium on Computer Architecture (ISCA), 2014.

[80] J. M. Sibigtroth et al. Memory bit line segment isolation. *U.S. patent*, Patent number 7042765, 2006.

[81] M. Stanisavljevic, H. Pozidis, A. Athmanathan, N. Papandreou, T. Mittelholzer, and E. Eleftheriou. Demonstration of reliable triple-level-cell (tlc) phase-change memory. In *2016 IEEE 8th International Memory Workshop (IMW)*, pages 1–4, 2016.

[82] J. Stuecheli, D. Kaseridis, D. Daly, H. C. Hunter, and L. K. John. The virtual write queue: Coordinating dram and last-level cache policies. International Symposium on Computer Architecture (ISCA), 2010.

[83] K. Sudan et al. *Micro-pages: Increasing DRAM eXciency with localityaware data placement.* ASPLOS, 2010.

[84] S. Swami and K. Mohanram. Adam: Architecture for write disturbance mitigation in scaled phase change memory. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1235–1240, 2018.

[85] M. K. Tavana and D. Kaeli. Cost-effective write disturbance mitigation techniques for advancing pcm density. pages 253–260. 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), November 2017.

[86] I. G. Thakkar and S. Pasricha. 3d-prowiz: An energy-efficient and optically-interfaced 3d dram architecture with reduced data access overhead. *IEEE Transactions on Multi-Scale Computing Systems*, 1(3):168–184, 2015.

[87] I. G. Thakkar and S. Pasricha. Dyphase: A dynamic phase change memory architecture with symmetric write latency and restorable endurance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(9):1760–1773, 2018.

[88] S. Thoziyoor et al. *A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies*. ISCA, 2008.

[89] C. Toal, D. Burns, K. McLaughlin, S. Sezer, and S. O'Kane. An rldram ii implementation of a 10gbps shared packet buffer for network processing. In *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, pages 613–618, 2007.

[90] R. Wang, L. Jiang, Y. Zhang, and J. Yang. Sd-pcm: Constructing reliable super dense phase change memory under write disturbance. *SIGARCH Comput. Archit. News*, 43(1):19–31, Mar. 2015.

[91] Z. Wang, S. Shan, T. Cao, J. Gu, Y. Xu, S. Mu, Y. Xie, and D. A. Jiménez. Wade: Writeback-aware dynamic cache management for nvm-based main memory system. *ACM Transactions on Architecture and Code Optimization (TACO)*, 10(4):2013, 2013.

[92] F. A. Ware and C. Hampel. Improving power and data efficiency with threaded memory modules. In *2006 International Conference on Computer Design*, pages 417–424, 2006.

[93] W. A. Wong and J.-L. Baer. Dram caching. Technical report, UW-CSE97-03-04, 1997.

[94] W. A. Wulf and S. A. McKee. Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24, Mar. 1995.

[95] F. Xia, D. Jiang, J. Xiong, M. Chen, L. Zhang, and N. Sun. Dwc: Dynamic write consolidation for phase change memory systems. International Conference on Supercomputing (ICS), 2014.

[96] H. Yoon, J. Meza, N. Muralimanohar, N. P. Jouppi, and O. Mutlu. Efficient data mapping and buffering techniques for multilevel cell phase-change memories. *ACM Transactions on Architecture and Code Optimization (TACO)*, 11(4):2015, 2015.

[97] J. Yue and Y. Zhu. Making write less blocking for read accesses in phase change memory. In *Proc. MASCOTS*, pages 269–277, 2012.

[98] J. Yue and Y. Zhu. Accelerating write by exploiting pcm asymmetries. In *Proc. HPCA*, pages 282–293, 2013.

[99] J. Yue and Y. Zhu. *Accelerating write by exploiting PCM asymmetries*. In High Performance Computer Architecture (HPCA), 2013.

[100] D. Zhang, L. Ju, M. Zhao, X. Gao, and Z. Jia. Write-back aware shared last-level cache management for hybrid main memory. In *Proc. DAC*, pages 1–6, 2016.

[101] L. Zhang, B. Neely, D. Franklin, D. Strukov, Y. Xie, and F. T. Chong. Mellow writes: Extending lifetime in resistive memories through selective slow write backs. International Symposium on Computer Architecture (ISCA), 2016.

[102] Z. Zhang, Z. Zhu, and X. Zhang. Cached dram for ilp processor memory access latency reduction. *IEEE Micro*, 21(4):22–32, 2001.

[103] H. Zheng, J. Lin, Z. Zhang, E. Gorbatov, H. David, and Z. Zhu. Mini-rank: Adaptive dram architecture for improving memory power efficiency. In *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pages 210–221, 2008.

[104] W. Zhou, D. Feng, Y. Hua, J. Liu, F. Huang, and Y. Chen. An efficient parallel scheduling scheme on multi-partition pcm architecture. International Symposium on Low Power Electronics and Design (ISLPED), 2016.

Vita

Chao-Hsuan Huang

Place of Birth:

- Changhua, Taiwan

Education:

- University of Kentucky, Lexington, Kentucky
  M.A. in Engineering, Dec. 2021
- National Chi Nan University, Nantou, Taiwan
  B.S. in Science, June. 2017

Honors

- ESWEEK 2020 travel grant
- NSF IGSC 2020 Student Participation Support

Publications & Preprints:

- Chao-Hsuan Huang and Ishan G Thakkar, "Improving the Latency-Area Trade-offs for DRAM Design with Coarse-Grained Monolithic 3D (M3D) Integration," 2020 IEEE 38th International Conference on Computer Design (ICCD), 2020
- Chao-Hsuan Huang and Ishan G Thakkar. 2019. Mitigating write disturbance in phase change memory architectures: work-in-progress. In Proceedings of the International Conference on Compliers, Architectures and Synthesis for Embedded Systems Companion (CASES '19). Association for Computing Machinery(ACM), New York, NY, USA, Article 7, 1–2.